Stationary Processes for Object Detection and Non-Rigid Structure-from-Motion

Jack Valmadre

Submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy.

School of Electrical Engineering and Computer Science
Science and Engineering Faculty
Queensland University of Technology
2016

Abstract

Stationary processes are random variables whose value is a signal and whose distribution is invariant to translation in the domain of the signal. They are intimately connected to convolution, and therefore to the Fourier transform, since the covariance matrix of a stationary process is a Toeplitz matrix, and Toeplitz matrices are the expression of convolution as a linear operator. This thesis utilises this connection in the study of i) efficient training algorithms for object detection and ii) trajectory-based non-rigid structure-from-motion.

Object detection is the problem of finding all instances of an object class in a photograph. When formulated as sliding-window classification, it is generally considered infeasible to train the classifier on the entire negative set. This thesis investigates two recent algorithms for efficiently training a binary classifier using a large set of negative images: Stationary Process Linear Discriminant Analysis (SPLDA) and Correlation Filters. Whereas SPLDA imposes the assumption that natural images are a stationary process by adopting a Toeplitz covariance matrix, Correlation Filters are trained using the Fourier transform of a subset of negative examples.

The two algorithms are shown to be equivalent except for their estimation of the covariance matrix. The effect of this difference is that the linear equation defining a Correlation Filter is much easier to solve, while the covariance matrix in SPLDA does not depend on the window size and only

needs to be computed once ever. A unified pipeline is conceived whereby the linear equation for a Correlation Filter can be constructed easily from the SPLDA covariance matrix, and the solution of the SPLDA equation is accelerated. The two algorithms are rigorously compared using two standard pedestrian detection datasets.

Non-Rigid Structure-from-Motion (NRSfM) is the problem of estimating the 3D structure of a deforming object from a single 2D projection per instant. This is a severely under-constrained problem, for which it is necessary to introduce external constraints. A popular approach is to impose the constraint that the trajectory of every point can be represented as a linear combination of low-frequency Discrete Cosine Transform (DCT) basis vectors.

This thesis shows that adopting a DCT basis is equivalent to imposing a stationary prior on the symmetric extension of the trajectory, and that the alternative use of a convolutional prior with compact support obviates the need to specify the basis dimension. The existing heuristic for reconstruction quality is replaced with a theoretical bound that justifies this decision. The use of a compact filter to encourage smoothness is further shown to admit a vastly more efficient solution to the reconstruction of articulated trajectories using dynamic programming.

Contents

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

QUT Verified Signature

Jack Valmadre

17/03/16

Date

Acknowledgements

Many people contributed to making my introduction to research a mostly enjoyable and thoroughly educational experience.

Primarily I thank Simon Lucey for the balance of guidance and freedom that he has given me since I was an undergraduate intern at CSIRO. I'm grateful to have had a supervisor who was excited by the theory as much as the applications. I also extend my gratitude to Sridha Sridharan for his support throughout and for handling the administrative aspects of my PhD.

I greatly appreciate my friends and mentors from my time at CSIRO. Mark Cox for the virtue that he brings to his research, his ever-rational opinions, his selfless advice and for echoing my desire to fully understand things. Jesus Nuevo Chiquero for his positivity, for promoting balance, for his wide knowledge of practical computer vision and his fresh bread. Jason Saragih for demonstrating ultimate pragmatism and speed, for his probing intellect and wry humour.

I'm lucky to have started and be finishing my PhD at the same time as Hilton Bristow. It was so often a relief to have a great friend on the same journey, especially one who prizes knowledge, rationality, integrity and side-projects so highly.

Thanks to Corina Gurău for her encouragement and enthusiasm, and for inspiring and challenging me.

I'm indebted to Yaser Sheikh for enabling me to spend an eye-opening semester at Carnegie Mellon University and for welcoming me into his group.

My work would not have been possible without the financial support of Queensland University of Technology and the Australian Research Council. I'm also especially appreciative of the two distributed computing environments that were made available to me. First the CI2CV cluster in the SAIVT lab, which was enabled by significant effort on the part of Mark Cox, and later the QUT High-Performance Computing and Research Support Group.

To name a few more friends who had a great impact... Daniel, Ella and Bec for being a constant positive presence in my life and giving me the perfect early-twenties-shared-Queenslander-in-Brisbane experience. Anouska, Courtney, Adam, Tim and Georgie for being great friends when I needed it. Claude and Evelyne for giving me a memorably fun start at CSIRO. Yingying for being contagiously inquisitive of the natural world. Ash, Kit, Iman and Chen for giving me a chance to be a mentor myself. Lisa and Emma for making my first trip to Pittsburgh fantastic. Shaurya, Varun, Daniel and Dey at CMU for discussions of research and everything else. Eléonore for various two-wheeled adventures in Pittsburgh.

Finally, I give profound thanks to my parents, Malcolm and Wendy, who gave me everything. I hope you don't miss me too much while I'm away!

Chapter 1

Introduction

One of the fundamental aspects that differentiates computer vision from pure machine learning is the ubiquity of signals: the data are not unstructured vectors, but sampled functions of space and/or time. Machine learning is intrinsically a study of probability distributions, and the theory that describes distributions of functions is that of random processes. Stationary processes are a subset of random processes whose distributions are invariant to translation in the domain of the signal, a property of many natural signals.

Convolution is a signal processing operation that computes the inner product of two signals at all possible relative translations. Stationary processes are inherently linked to convolution through Toeplitz operators: the covariance matrix of a stationary process is a Toeplitz operator, and the linear operator that represents convolution with a signal is a Toeplitz operator.

This thesis explores the use of stationary processes, Toeplitz operators and convolution in two seemingly disparate problems in computer vision: object detection and non-rigid structure-from-motion.

1.1 Object Detection

Object detection is the problem of finding all instances of an object class in a photograph. Two common examples with clear practical applications are the detection of pedestrians and cars in street scenes. Rather than attempt the manual design of such a detector, the modern approach is to develop an algorithm that learns to recognise the class from a set of examples. A simple way to construct an object detector is to frame detection as the binary classification of all windows in an image. The data are images and their distribution can be modelled as a two-dimensional stationary process.

This thesis will compare and contrast two methods that adopt this assumption, one for statistical and one for computational reasons. The two methods are unified into a single framework.

Fast and lightweight algorithms for training an object detector could be useful for adaptive visual tracking (especially in embedded systems), for image search by visual query, or more generally to learn higher-level functions of images that are composed of linear detectors.

1.2 Non-Rigid Structure-from-Motion

Structure-from-Motion (SfM) is the problem of estimating the 3D structure of an object from several 2D projections that capture different views. The canonical problem assumes that a single shape is observed by cameras in different positions, or equivalently that the object undergoes a rigid transformation between images. A rigid transform preserves distances between points, and comprises just translation and rotation.

Non-Rigid Structure-from-Motion (NRSfM) is the more general problem in which the object is able to deform between observations. This is a severely

under-constrained problem, for which it is necessary to introduce external constraints. The standard approach is to impose the constraint that the set of 3D shapes is linearly dependent and has low rank, meaning that every shape can be represented as a weighted combination of a small number of basis shapes. It was later recognised that there is a dual interpretation of this property: that the trajectory of every point can be represented as a weighted combination of the same number of basis trajectories. The advantage of a trajectory basis is that it does not need to be learnt since any basis that promotes smooth motion can be used. Past approaches have adopted a truncated Discrete Cosine Transform (DCT) basis for its documented ability to efficiently represent natural signals.

This thesis shows that adopting a DCT basis is equivalent to imposing a stationary prior on the symmetric extension of the trajectory, however the alternative use of a convolutional prior with compact support obviates the need to specify the basis dimension. The existing heuristic for reconstruction quality is replaced with a theoretical bound that justifies this decision. The use of a compact filter to encourage smoothness is further shown to admit a vastly more efficient solution to the reconstruction of articulated trajectories using dynamic programming.

1.3 Publications

The contributions of Chapter 4 appeared in the conference paper

- *"Learning detectors quickly with stationary statistics,"* Jack Valmadre, Sridha Sridharan and Simon Lucey, ACCV 2014,

and the techniques of that paper constituted an important part of the paper

- *"Dense semantic correspondence where every pixel is a classifier,"* Hilton Bristow, Jack Valmadre and Simon Lucey, ICCV 2015.

The research that comprises Chapter 6 culminated in the publications

- *"General trajectory prior for non-rigid reconstruction,"* Jack Valmadre and Simon Lucey, CVPR 2012, and
- *"Efficient articulated trajectory reconstruction using dynamic programming and filters,"* Jack Valmadre, Yingying Zhu, Sridha Sridharan and Simon Lucey, ECCV 2012.

Additionally, the following publication arose from a collaboration with Carnegie Mellon University, although its content matter does not appear in this thesis

- *"Separable spatiotemporal priors for convex reconstruction of time-varying 3D point clouds,"* Tomas Simon, Jack Valmadre, Iain Matthews and Yaser Sheikh, ECCV 2014.

Chapter 2

Preliminaries

2.1 Notation

Where possible, operators (matrices) are assigned upper-case symbols, while vectors and signals are assigned lower-case symbols. Blackboard-style (double-barred) symbols are use for canonical sets, and calligraphic symbols for other sets. Common operations and sets are outlined in Table 2.1.

The addition of a set and an element is defined

$$\mathcal{A} + x = \{a + x : a \in \mathcal{A}\} = \{a : a - x \in \mathcal{A}\} \ . \tag{2.1}$$

The addition of two sets is the Minkowski sum

$$\mathcal{A} + \mathcal{B} = \{a + b : a \in \mathcal{A}, b \in \mathcal{B}\} \tag{2.2}$$

and the difference of two sets $\mathcal{A} - \mathcal{B}$ is the analogous Minkowski difference.

The p-norm of a vector $x = (x_1, \ldots, x_n)$ with $p \in \mathbb{R}$ and $p \geq 1$ is denoted

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{2.3}$$

Symbol	Meaning
$\mathbb{Z}$	the set of integers
$\mathbb{R}$	the set of real numbers
$\mathbb{C}$	the set of complex numbers
$\mathbb{Z}_n$	$\{u \in \mathbb{Z}^d : 0 \leq u < n\}$ for $n \in \mathbb{Z}^d$
$*$	convolution
$\star$	cross-correlation
$\odot$	element-wise product
$\otimes$	Kronecker product
F	Fourier transform
$\hat{x}$	transform of x (usually Fourier)
x^*	complex conjugate of x
x^H	conjugate transpose of x
$A^\dagger$	pseudo-inverse of A

Table 2.1: Notation for common operations and sets.

with the special case $p = \infty$ defined

$$\|x\|_\infty = \max_i |x_i| \ . \tag{2.4}$$

Unless otherwise stated, the norm of a vector refers to its Euclidean norm $\|x\| = \|x\|_2$.

The shorthand $\|x\|_A = \sqrt{x^T A x}$ is introduced for positive semidefinite operators $A \succeq 0$, although it is not strictly a norm unless A is positive definite $A \succ 0$.

The norm of an operator A is the norm induced by that vector norm

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\| \leq 1} \|Ax\| \ . \tag{2.5}$$

The 2-norm of a matrix, also known as the spectral norm, is its maximum singular value $\|A\|_2 = \sigma_{\max}(A)$.

Vector inequalities imply the scalar inequality of all elements. For example, if a and b are vectors in $\mathbb{R}^n$ with elements $a_1, \ldots, a_n$ and $b_1, \ldots b_n$ respectively, then $a \leq b$ is shorthand for

$$a_t \leq b_t \quad \forall t = 1, \ldots, n \ . \tag{2.6}$$

The modulo operation has the lowest precedence. For example, $a + b \bmod \ell = (a + b) \bmod \ell$ and $-a \bmod \ell = (-a) \bmod \ell$. When applied to vectors of the same dimension, it acts element-wise

$$a \bmod \ell = (a_1, \ldots, a_n) \bmod (\ell_1, \ldots, \ell_n) = (a_1 \bmod \ell_1, \ldots, a_n \bmod \ell_n) \ . \tag{2.7}$$

2.2 Signals

A scalar-valued discrete signal x is a map $x : \mathcal{U} \to \mathbb{R}$ that assigns a real number $x[u]$ to every element u in its index space $\mathcal{U}$. The index space of a

d-dimensional signal is a d-dimensional Cartesian grid $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_d$ where each $\mathcal{U}_i \subseteq \mathbb{Z}$ is a set of consecutive integers. The elements of the index space are d-tuples $u = (u_1, \ldots, u_d) \in \mathcal{U}$.

Scalar-valued discrete signals are useful for representing sampled continuous quantities such as mono-channel audio and greyscale images. However, to represent richer sources of information such as stereo-channel audio and color images, it is necessary to introduce multi-channel or vector-valued discrete signals. A multi-channel discrete signal with k channels is a map $x : \mathcal{U} \to \mathbb{R}^k$ that instead assigns a real vector $x[u] = (x_1[u], \ldots, x_k[u]) \in \mathbb{R}^k$ to each element $u \in \mathcal{U}$. Subscripts are used to denote channel indices and square brackets are used to denote elements of the index space. This enables the notation $x_p : \mathcal{U} \to \mathbb{R}$ that refers to channel p alone as a scalar-valued signal.

If a signal is finite, then its size $\ell = (\ell_1, \ldots, \ell_d) \in \mathbb{Z}^d$ specifies the number of elements in each dimension of the index space $\ell_i = |\mathcal{U}_i|$. Most commonly, a signal of size ℓ will have index space $\mathcal{U} = \mathbb{Z}_\ell$.

A finite signal with k channels and $m = |\mathcal{U}| = \prod_i \ell_i$ elements in its domain can be considered a real vector with mk scalar elements that are indexed $x_p[u] \in \mathbb{R}$ for $(u, p) \in \mathcal{J} = \mathcal{U} \times \{1, \ldots, k\}$ instead of $x_i \in \mathbb{R}$ for $i \in \{1, \ldots, mk\}$. More generally, the set of k-channel signals with (possibly infinite) domain $\mathcal{U}$ is a vector space with inner product

$$x^T y = \sum_{(u,p) \in \mathcal{J}} x_p[u]\, y_p[u] \tag{2.8}$$

using the familiar transpose notation for column vectors.

2.3 Linear Operators for Signals

A function A is a linear operator if it satisfies $A(x + y) = A(x) + A(y)$ and $A(\alpha x) = \alpha A(x)$ for any vectors x, y and scalar α [24]. The evaluation of a

linear operator $A(x)$ will be abbreviated Ax.

Any linear operator A that maps vectors in $\mathbb{R}^n$ to vectors in $\mathbb{R}^{n'}$ has a representation as multiplication by an $n' \times n$ matrix with elements $A_{ij} \in \mathbb{R}$

$$(Ax)_i = \sum_{j=1}^{n} A_{ij} x_j \qquad i = 1, \ldots, n' \ . \tag{2.9}$$

Similarly, a linear operator A that maps k-channel signals with domain $\mathcal{U}$ to k'-channel signals with domain $\mathcal{U}'$ has a representation as multiplication by a matrix whose rows correspond to $(u, p) \in \mathcal{J}' = \mathcal{U}' \times \{1, \ldots, k'\}$ and columns correspond to $(t, q) \in \mathcal{J} = \mathcal{U} \times \{1, \ldots, k\}$. To preserve the signal structure in the array that defines the operator, this thesis introduces the notation that its elements are indexed $A_{pq}[u, t] \in \mathbb{R}$ for $p \in \{1, \ldots, k'\}$, $q \in \{1, \ldots, k\}$, $u \in \mathcal{U}'$ and $t \in \mathcal{U}$. Subscripts are again used for channel indices, and square brackets for positions in the signal domain. The action of a linear operator is

$$(Ax)_p[u] = \sum_{(t,q) \in \mathcal{J}} A_{pq}[u, t]\, x_q[t] \qquad (u, p) \in \mathcal{J}' \ . \tag{2.10}$$

A linear operator that maps k-channel signals of size $\ell = (\ell_1, \ldots, \ell_d) \in \mathbb{Z}^d$ to k'-channel signals of size $\ell' = (\ell'_1, \ldots, \ell'_{d'}) \in \mathbb{Z}^{d'}$ is therefore defined by an array with $d + d' + 2$ dimensions $k' \times k \times \ell'_1 \times \cdots \times \ell'_{d'} \times \ell_1 \times \cdots \times \ell_d$ instead of a matrix with two dimensions $(k' \ell'_1 \cdots \ell'_{d'}) \times (k \ell_1 \cdots \ell_d)$.

This notation enables A_{pq} and $A[u, t]$ to denote the appropriate sub-matrices such that A_{pq} is a linear operator that maps scalar-valued signals to scalar-valued signals and $A[u, t]$ is a linear operator that maps vectors in $\mathbb{R}^k$ to vectors in $\mathbb{R}^{k'}$. The product Ax can equivalently be expressed as a sum of channel-wise operators

$$(Ax)_p = \sum_{q=1}^{k} A_{pq}\, x_q \qquad\qquad p = 1, \ldots, k' \tag{2.11}$$

or a sum of sample-wise operators

$$(Ax)[u] = \sum_{t \in \mathcal{U}} A[u,t]\, x[t] \qquad u \in \mathcal{U}' \ . \tag{2.12}$$

The transpose of a linear operator is obtained by exchanging the indices of the row (u,p) and column (t,q). Therefore if $B = A^T$, then $B_{pq}[u,t] = A_{qp}[t,u]$, and consequently

$$B_{pq} = (A_{qp})^T \qquad\qquad B[u,t] = (A[t,u])^T \ . \tag{2.13}$$

Similar to the expression for the inner product of two signals, the outer product $A = xy^T$ is a square, rank-one operator with elements

$$A_{pq}[u,t] = x_p[u]\, y_q[t] \qquad (u,p),(t,q) \in \mathcal{J} \tag{2.14}$$

such that $(xy^T)z = x(y^Tz)$ where z is another signal of the same class.

2.4 Random Processes

A random process (otherwise known as a stochastic process or random field) is a collection of random variables $X[u]$ whose indices u are elements of an index space $\mathcal{U}$ with $d \geq 1$ dimensions [1]. The random variable at each index may itself be a random vector with $k \geq 1$ elements $X[u] = (X_1[u], \ldots, X_k[u])$. While the index space can in general be real-valued, this work will only consider integer spaces, where the distribution of a random process is a distribution over discrete signals on the same domain. The mean of a random process is a signal $\bar{x} : \mathcal{U} \to \mathbb{R}^k$. The covariance matrix of a random process is a linear operator that maps signals to signals as described in the previous section, defined as the expectation of outer products

$$S = \mathbb{E}\left\{(X - \bar{x})(X - \bar{x})^T\right\} \ . \tag{2.15}$$

2.5 Convolution

Convolution is an operation that computes the inner product of two signals at all relative shifts. The convolution of two signals $x : \mathbb{Z} \to \mathbb{R}$ and $y : \mathbb{Z} \to \mathbb{R}$ is a signal $x * y : \mathbb{Z} \to \mathbb{R}$ defined [48]

$$(x * y)[u] = \sum_{t \in \mathbb{Z}} x[t]\, y[u - t] \qquad u \in \mathbb{Z} \ . \tag{2.16}$$

More generally, the convolution of two d-dimensional signals $x : \mathbb{Z}^d \to \mathbb{R}$ and $y : \mathbb{Z}^d \to \mathbb{R}$ is a d-dimensional signal $x * y : \mathbb{Z}^d \to \mathbb{R}$ defined

$$(x * y)[u] = \sum_{t \in \mathbb{Z}^d} x[t]\, y[u - t] \qquad u \in \mathbb{Z}^d \ . \tag{2.17}$$

Convolution is commutative $y * x = x * y$ since

$$(y * x)[u] = \sum_{t \in \mathbb{Z}^d} y[t]\, x[u - t] = \sum_{\tau \in \mathbb{Z}^d} y[u - \tau]\, x[\tau] = (x * y)[u] \tag{2.18}$$

using the fact that $\tau = u - t$ is a bijective map for $t, \tau \in \mathbb{Z}^d$.

2.6 Toeplitz Operators

Toeplitz matrices are matrices with constant diagonals [24]. The elements $A[u, t]$ of a Toeplitz matrix A are specified by a one-dimensional array

$$A[u, t] = a[u - t] \ . \tag{2.19}$$

The form of an $\ell' \times \ell$ Toeplitz matrix is

$$A = \begin{bmatrix} a[0] & a[-1] & a[-2] & \cdots & a[-\ell + 1] \\ a[1] & a[0] & a[-1] & \cdots & a[-\ell + 2] \\ a[2] & a[1] & a[0] & \cdots & a[-\ell + 3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a[\ell' - 1] & a[\ell' - 2] & a[\ell' - 3] & \cdots & a[-\ell + \ell'] \end{bmatrix} . \tag{2.20}$$

Such a matrix is fully specified by the $\ell + \ell' - 1$ elements of its first row and column $a[\delta]$ for $-\ell < \delta < \ell'$. If A is square $\ell' = \ell$ and symmetric $A = A^T$, then $A[u,t] = a[|u-t|]$ and the matrix is defined by ℓ elements.

The one-dimensional convolution of one signal with another is a linear operator $Ax = x * a$ that has elements $A[u,t] = a[u-t]$ for $u, t \in \mathbb{Z}$ and is therefore Toeplitz. This class of Toeplitz operators, which are defined on the set of all integers, are known as bi-infinite Toeplitz operators or Laurent operators [24].

Finite Toeplitz operators compute a subset of an infinite convolution. If A is a finite Toeplitz operator of size $\ell' \times \ell$ that maps signals on $\mathcal{U} = \{0, \ldots, \ell-1\}$ to signals on $\mathcal{U}' = \{0, \ldots, \ell' - 1\}$ and has elements $A[u,t] = a[u-t]$, then

$$(Ax)[u] = \sum_{t \in \mathcal{U}} a[u-t]\, x[t] = (\tilde{x} * \tilde{a})[u] \qquad u \in \mathcal{U}' \tag{2.21}$$

where $\tilde{x} : \mathbb{Z} \to \mathbb{R}$ is the extension of x with zeros

$$\tilde{x}[u] = \begin{cases} x[u], & u \in \mathcal{U} \\ 0, & \text{otherwise} \end{cases} \tag{2.22}$$

and $\tilde{a} : \mathbb{Z} \to \mathbb{R}$ satisfies $\tilde{a}[u] = a[u]$ for $u \in \mathcal{U}' - \mathcal{U} = \{-\ell + 1, \ldots, \ell' - 1\}$.

Toeplitz operators are generalised to vector-valued signals in *block* Toeplitz operators. A block Toeplitz operator that maps k-channel signals to k'-channel signals has elements

$$A_{pq}[u,t] = a_{pq}[u-t] \tag{2.23}$$

for $p = 1, \ldots, k'$ and $q = 1, \ldots, k$. For infinite signals, these operators can be understood as convolution where $a[u-t]$ is no longer a scalar but a $k' \times k$ matrix

$$(Ax)[u] = \sum_{t \in \mathbb{Z}} a[u-t]\, x[t] \qquad u \in \mathbb{Z} \tag{2.24}$$

or alternatively as a sum of convolutions

$$(Ax)_p = \sum_{q=1}^{k} (a_{pq} * x_q) \qquad p = 1, \ldots, k' \ . \tag{2.25}$$

If a block Toeplitz matrix is symmetric $A = A^T$, then the array of unique elements has symmetry $a_{pq}[\delta] = a_{qp}[-\delta]$ since

$$a_{pq}[u - t] = A_{pq}[u, t] = A_{qp}[t, u] = a_{qp}[t - u] \ . \tag{2.26}$$

This can equivalently be expressed $a[\delta] = (a[-\delta])^T$, or $a_{pq} = J a_{qp}$ where J is the signal reversal operator $(Jx)[u] = x[-u]$.

The $k'\ell' \times k\ell$ matrix that corresponds to a block Toeplitz operator from k-channel signals of length ℓ to k'-channel signals of length ℓ' may be a Toeplitz matrix of unstructured blocks, or an unstructured matrix of Toeplitz blocks. This difference is simply a permutation of the vector indices.

2.7 Multi-Level Toeplitz Operators

The class of linear operators that correspond to multi-dimensional convolution are called *multi-level* (or *multi-index*) Toeplitz operators [64], and their extension to vector-valued signals is in *block multi-level* Toeplitz operators. A linear operator A that maps k-channel signals with domain $\mathcal{U} \subseteq \mathbb{Z}^d$ to k'-channel signals with domain $\mathcal{U}' \subseteq \mathbb{Z}^d$ is block multi-level Toeplitz if and only if its elements can be specified by a lower-dimensional array $A_{pq}[u, t] = a_{pq}[u - t]$, or more explicitly

$$A_{pq}[(u_1, \ldots, u_d), (t_1, \ldots, t_d)] = a_{pq}[(u_1 - t_1, \ldots, u_d - t_d)] \ . \tag{2.27}$$

It therefore computes

$$(Ax)_p[u] = \sum_{t \in \mathcal{U}} \sum_{q=1}^{k} a_{pq}[u - t] x_q[t] \tag{2.28}$$

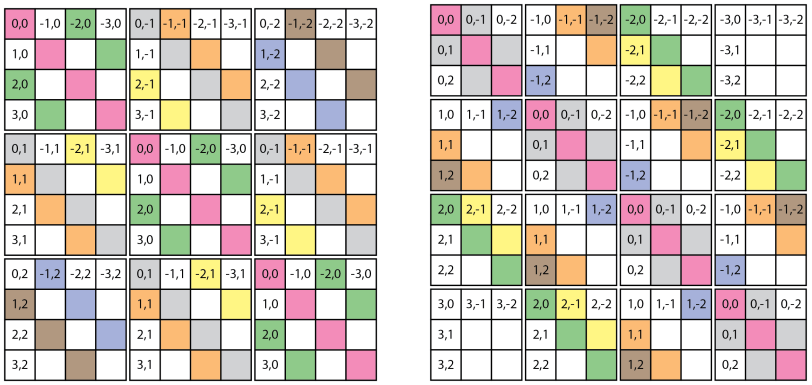


Figure 2.1: The 12×12 matrix representation of a symmetric two-level Toeplitz operator for signals of size 4×3 may be arranged as a 3×3 Toeplitz array of 4×4 Toeplitz blocks, or alternatively as a 4×4 Toeplitz array of 3×3 Toeplitz blocks. Neither is a one-level Toeplitz matrix.

for $u \in \mathcal{U}'$ and $p = 1, \ldots, k'$. The input and output signals must have identical dimension d and the array of unique elements $a_{pq}[\delta]$ needs to be defined for $\delta = (\delta_1, \ldots, \delta_d) \in \mathcal{U}' - \mathcal{U}$. For a block multi-level Toeplitz operator that maps signals of size $\ell = (\ell_1, \ldots, \ell_d)$ to signals of size $\ell' = (\ell'_1, \ldots, \ell'_d)$, this array has $d + 2$ dimensions $k' \times k \times (\ell_1 + \ell'_1 - 1) \times \cdots \times (\ell_d + \ell'_d - 1)$. This array has far fewer elements than there are in the full matrix.

As was the case for block (one-level) Toeplitz operators, the structure of the matrix that represents a multi-level Toeplitz operator depends on the specific order in which multi-dimensional signals are vectorised. Figure 2.1 depicts that the $\ell_1\ell_2 \times \ell_1\ell_2$ matrix of a two-level Toeplitz operator with an input and output domain of size $\ell = (\ell_1, \ell_2)$ may be an $\ell_1 \times \ell_1$ Toeplitz matrix of $\ell_2 \times \ell_2$ Toeplitz matrix blocks, or an $\ell_2 \times \ell_2$ Toeplitz matrix of $\ell_1 \times \ell_1$ Toeplitz matrix blocks.

2.8 Stationary Processes

A random process $X[u]$ with index space $u \in \mathcal{U}$ is said to be stationary if its distribution is invariant to translation [1]. That is, any subset of random variables $\mathcal{A} = \{u_1, \ldots, u_n\} \subseteq \mathcal{U}$ of any size n has the same distribution as the shifted subset $\mathcal{A} + \tau$ for any shift τ such that $\mathcal{A} + \tau \subseteq \mathcal{U}$. Formally, the cumulative distribution function (cdf)

$$F_{\{u_1,\ldots,u_n\}}(\alpha_1, \ldots, \alpha_n) = \Pr(X[u_1] \leq \alpha_1, \ldots, X[u_n] \leq \alpha_n) \tag{2.29}$$

is unchanged $F_{\mathcal{A}}(\alpha) = F_{\mathcal{A}+\tau}(\alpha)$ for all valid τ. This definition encompasses k-channel signals where each $\alpha_i \in \mathbb{R}^k$.

If a random process is stationary, then translation invariance implies that its mean $\bar{x} : \mathcal{U} \to \mathbb{R}^k$ is a uniform signal

$$\bar{x}[u] = \mathbb{E}\{X[u]\} = \bar{x}[u + \tau] = \mu \tag{2.30}$$

where $\mu = (\mu_1, \ldots, \mu_k) \in \mathbb{R}^k$ is a single vector-valued sample. Translation invariance further implies that the covariance of a stationary process is a block multi-level Toeplitz operator where the covariance of $X[u]$ and $X[t]$ is determined by their relative position alone

$$S[u, t] = \mathbb{E}\{(X[u] - \mu)(X[t] - \mu)^T\} = S[u + \tau, t + \tau] = s[u - t] \ . \tag{2.31}$$

2.9 Periodic Convolution

The *circular* or *periodic* convolution $x * y$ of two one-dimensional signals $x : \mathbb{Z} \to \mathbb{R}$ and $y : \mathbb{Z} \to \mathbb{R}$ that are periodic with identical period $x[u] = x[u + \ell]$ and $y[u] = y[u + \ell]$ is a signal $x * y : \mathbb{Z} \to \mathbb{R}$ defined [48]

$$(x * y)[u] = \sum_{t=0}^{\ell-1} x[t]\, y[u - t] \qquad u \in \mathbb{Z} \ . \tag{2.32}$$

It is periodic with same period as its inputs $(x * y)[u] = (x * y)[u + \ell]$.

More generally, the circular convolution of two d-dimensional periodic signals $x : \mathbb{Z}^d \to \mathbb{R}$ and $y : \mathbb{Z}^d \to \mathbb{R}$ with identical period $\ell = (\ell_1, \ldots, \ell_d)$ is a d-dimensional signal $x * y : \mathbb{Z}^d \to \mathbb{R}$ defined [43]

$$(x * y)[u] = \sum_{t \in \mathbb{Z}_\ell} x[t]\, y[u - t] \qquad u \in \mathbb{Z}^d \ . \tag{2.33}$$

It also inherits the period of its inputs $(x * y)[u] = (x * y)[u \bmod \ell]$.

Since a signal with period ℓ is fully defined by the values that it takes on one period, circular convolution can be considered a transform that maps two finite signals $x : \mathbb{Z}_\ell \to \mathbb{R}$ and $y : \mathbb{Z}_\ell \to \mathbb{R}$ to a finite signal $x * y : \mathbb{Z}_\ell \to \mathbb{R}$

$$(x * y)[u] = \sum_{t \in \mathbb{Z}_\ell} x[t]\, y[u - t \bmod \ell] \qquad u \in \mathbb{Z}_\ell \tag{2.34}$$

such that if $\tilde{x}[u] = x[u \bmod \ell]$ and $\tilde{y}[u] = y[u \bmod \ell]$ are the periodic extensions of x and y, then $(\tilde{x} * \tilde{y})[u] = (x * y)[u \bmod \ell]$.

2.10 Circulant Operators

Circulant Toeplitz operators (often simply "circulant operators") are the linear operators that compute the *circular* convolution of one signal with another. The circulant matrices are a subset of the finite square Toeplitz matrices. For a positive integer ℓ, the elements of an $\ell \times \ell$ circulant matrix A satisfy [28]

$$A[u, t] = a[u - t \bmod \ell] \tag{2.35}$$

with $u, t \in \{0, \ldots, \ell - 1\}$. Therefore the operator computes the circular convolution $Ax = x * a$ of one-dimensional signals $x : \{0, \ldots, \ell - 1\} \to \mathbb{R}$ and

$a : \{0, \ldots, \ell - 1\} \to \mathbb{R}$. The matrix A takes the form

$$A = \begin{bmatrix} a[0] & a[\ell-1] & a[\ell-2] & \cdots & a[1] \\ a[1] & a[0] & a[\ell-1] & \cdots & a[2] \\ a[2] & a[1] & a[0] & \cdots & a[3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a[\ell-1] & a[\ell-2] & a[\ell-3] & \cdots & a[0] \end{bmatrix} \tag{2.36}$$

and is therefore specified by $a[\delta]$ for $\delta \in \{0, \ldots, \ell - 1\}$, a total of ℓ elements. If the matrix is symmetric, then $\lceil (\ell + 1)/2 \rceil$ elements are sufficient.

More generally, the circular convolution of two signals with dimension $d \geq 1$, size $\ell = (\ell_1, \ldots \ell_d)$ and domain $\mathbb{Z}_\ell$ corresponds to a *multi-level* circulant operator $Ax = x * a$ that satisfies $A[u, t] = a[u - t \bmod \ell]$ where $u = (u_1, \ldots, u_d)$ and $t = (t_1, \ldots, t_d)$ are multi-dimensional indices [64, 43].

Periodic convolution, like infinite convolution, is commutative $x * y = y * x$ and associative $x * (y * z) = (x * y) * z$ [48]. This bestows circulant operators with useful properties that do not hold for general finite Toeplitz operators:

- The product of two circulant operators AB is circulant since $ABx = a * (b * x) = (a * b) * x$.

- Circulant operators commute $AB = BA$ since $ABx = a * (b * x) = b * (a * x) = BAx$.

2.11 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a linear operator F that maps one periodic signal $x : \mathbb{Z} \to \mathbb{C}$ to another $Fx : \mathbb{Z} \to \mathbb{C}$. For an input signal with

period $x[u] = x[u + \ell]$, the DFT is defined [48]

$$(Fx)[u] = \sum_{t=0}^{\ell-1} x[t] e^{-i2\pi ut/\ell} \qquad u \in \mathbb{Z} \tag{2.37}$$

and has the same period $(Fx)[u] = (Fx)[u + \ell]$. The inverse DFT of a periodic signal $x[u] = x[u + \ell]$ is similarly defined

$$(F^{-1}x)[u] = \frac{1}{\ell} \sum_{t=0}^{\ell-1} x[t] e^{i2\pi ut/\ell} \qquad u \in \mathbb{Z} \tag{2.38}$$

and also has the same period $(F^{-1}x)[u] = (F^{-1}x)[u + \ell]$.

More generally, the multi-dimensional DFT of a signal with period $\ell = (\ell_1, \ldots, \ell_d)$ is defined [43]

$$(Fx)[u] = \sum_{t \in \mathbb{Z}_\ell} x[t] \prod_{i=1}^{d} e^{-i2\pi u_i t_i/\ell_i} \qquad u \in \mathbb{Z}^d \tag{2.39}$$

with inverse

$$(F^{-1}x)[u] = \frac{1}{\ell_1 \cdots \ell_d} \sum_{t \in \mathbb{Z}_\ell} x[t] \prod_{i=1}^{d} e^{i2\pi u_i t_i/\ell_i} \qquad u \in \mathbb{Z}^d \tag{2.40}$$

and both have period ℓ.

Since periodic signals are sufficiently described in the values that they take on one period, the DFT can be considered a finite transform that maps signals with domain $\mathbb{Z}_\ell$ to signals with the same domain. The elements of the transform are

$$F[u, t] = \prod_{i=1}^{d} e^{-i2\pi u_i t_i/\ell_i} \qquad u, t \in \mathbb{Z}_\ell \tag{2.41}$$

and the finite inverse transform is $F^{-1} = \frac{1}{m} F^H$ where $m = \ell_1 \cdots \ell_2$ is the number of samples in one period. The DFT is an orthogonal transform $FF^H = F^H F = mI$ [48].

The DFT is of fundamental importance in signal processing due to the convolution theorem, which states that periodic convolution is equivalent to element-wise multiplication in the Fourier domain [48]

$$F(x * y) = (Fx) \odot (Fy) \ . \tag{2.42}$$

An alternative statement of this property is that the (multi-dimensional) DFT basis is a set of orthogonal eigenvectors for all (multi-level) circulant matrices [28, 64] with any matrix A that has elements $A[u, t] = a[u - t \bmod \ell]$ decomposed

$$A = F^{-1} \operatorname{diag}(Fa) F \ . \tag{2.43}$$

These statements of the convolution theorem are equivalent since

$$F(x * a) = FAx = FAF^{-1}Fx = \operatorname{diag}(Fa)Fx = (Fa) \odot (Fx) \ . \tag{2.44}$$

The property that all circulant matrices share a set of eigenvectors provides an alternative explanation for the properties that the product of two circulant matrices is circulant and that circulant matrices commute. It also further reveals that the inverse of a circulant matrix, if it exists, is circulant [28] since $A^{-1} = F^{-1} \operatorname{diag}(Fa)^{-1} F$.

2.12 Fast Fourier Transform

The Fast Fourier Transform (FFT) is an algorithm for computing the forward or inverse DFT of a signal with $m = \ell_1 \cdots \ell_d$ samples in $O(m \log m)$ time [48, 43]. This enables fast evaluation of (multi-level) circulant operators. The naive computation of a circulant operator for signals with $m = \ell_1 \cdots \ell_d$ samples as matrix multiplication takes $O(m^2)$ time. The FFT enables this to instead be obtained in $O(m \log m)$ time using Algorithm 2.1.

Algorithm 2.1 Fast evaluation of circulant operator $y = Ax = a * x$ for signals with dimension d, size $\ell = (\ell_1, \ldots, \ell_d)$ and $m = \ell_1 \ldots \ell_d$ elements.

Require: $a[u]$ and $x[u]$ for $u \in \mathbb{Z}_\ell$

Ensure: $y[u] = (Ax)[u]$ for $u \in \mathbb{Z}_\ell$

$\hat{x} \leftarrow Fx$ // $O(m \log m)$ time

$\hat{a} \leftarrow Fa$ // $O(m \log m)$ time

for all $u \in \mathbb{Z}_\ell$ **do**

$\hat{y}[u] \leftarrow \hat{a}[u]\, \hat{x}[u]$ // $O(m)$ time

end for

$y \leftarrow F^{-1}\hat{y}$ // $O(m \log m)$ time

2.13 Periodic Cross-Correlation

The circular (or periodic) cross-correlation $x \star y$ of two signals $x : \mathbb{Z}^d \to \mathbb{R}$ and $y : \mathbb{Z}^d \to \mathbb{R}$ that are periodic $x[u] = x[u \bmod \ell]$ and $y[u] = y[u \bmod \ell]$ with period $\ell = (\ell_1, \ldots, \ell_d)$ is defined similarly to circular convolution [26]

$$(x \star y)[u] = \sum_{t \in \mathbb{Z}_\ell} x[t]\, y[u + t] \qquad u \in \mathbb{Z}^d \ . \tag{2.45}$$

Let J denote the signal reversal operator $(Jx)[u] = x[-u]$. Cross-correlation is related to convolution $x \star y = (Jx) * y$ since

$$(x \star y)[u] = \sum_{t \in \mathbb{Z}_\ell} x[t]\, y[u + t] = \sum_{\tau \in \mathbb{Z}_\ell} x[-\tau]\, y[u - \tau] = ((Jx) * y)][u] \ . \tag{2.46}$$

It follows from the definition of the DFT that the reversal of a periodic signal corresponds to the conjugation of its transform $FJx = (Fx)^*$. Therefore cross-correlation can be performed in the Fourier domain

$$F(x \star y) = (Fx)^* \odot (Fy) \ . \tag{2.47}$$

This reveals that cross-correlation is not commutative. Exchanging the

operands results in a reversal of the cross-correlation $y \star x = J(x \star y)$ since

$$F(y \star x) = (Fy)^* \odot (Fx) = ((Fy) \odot (Fx)^*)^* = FJ(x \star y) \tag{2.48}$$

using the properties of complex numbers. When representing periodic signals with a finite signal on one period $\mathbb{Z}_\ell$, the above results hold using the reversal operator $(Jx)[u] = x[-u \bmod \ell]$ for $u \in \mathbb{Z}_\ell$.

If A is a (multi-level) circulant Toeplitz operator with elements $A[u,t] = a[u-t \bmod \ell]$ that therefore computes circular convolution $Ax = x*a = a*x$, then A^T is a (multi-level) circulant Toeplitz operator that computes circular cross-correlation $A^T x = a \star x$. This is because its elements are $(A^T)[u,t] = a[t-u \bmod \ell] = (Ja)[u-t \bmod \ell]$ and therefore $A^T x = x * (Ja) = a \star x$.

While the matrix that corresponds to *left* cross-correlation $Ax = a \star x$ is circulant Toeplitz with elements $A[u,t] = (Ja)[u-t \bmod \ell]$, the matrix that corresponds to *right* cross-correlation $Ax = x \star a$ is instead circulant *Hankel* [7] with elements $A[u,t] = a[u+t \bmod \ell]$. Circulant Hankel matrices are not diagonalised by the Fourier basis, however the product AB of two circulant Hankel matrices $A[u,t] = a[u+t \bmod \ell]$ and $B[u,t] = b[u+t \bmod \ell]$ is a circulant Toeplitz matrix since

$$F(ABx) = F((x \star a) \star b) = (\hat{x}^* \odot \hat{a})^* \odot \hat{b} = \hat{x} \odot \hat{a}^* \odot \hat{b} = F((a \star b) * x) \ . \tag{2.49}$$

2.14 Bi-infinite Toeplitz Operators

Bi-infinite Toeplitz operators are similar to circulant Toeplitz matrices in that every bi-infinite Toeplitz matrix is diagonalised by the *Discrete-Time* Fourier Transform (DTFT). It follows that these operators commute $ABx = BAx$, and that the inverse of an operator, if it exists, is also a bi-infinite Toeplitz operator [24]. An important ramification is that the precision matrix (inverse covariance matrix) of a stationary process with infinite extent is also Toeplitz.

Chapter 3

Efficient Training Algorithms for Object Detection

3.1 Problem Description

Given a photograph, visual object detection is the problem of identifying the instances of a physical class and their visible extent. Instance locations may be specified using, for example, a bounding box or pixel segmentation. It is, perhaps surprisingly to humans, extremely difficult to construct a system that has simultaneously a low rate of false positives (hallucinated detections) and false negatives (undetected instances). This is because the appearance of an object is the product of many factors, primarily within-class variation, rigid and non-rigid pose, and scene properties such as lighting and background (Figure 3.1). Images are high-dimensional, and the subset of images belonging to an object class has a complex geometry with many degrees of freedom. Further complications arise from partial occlusion by other objects and truncation at image boundaries, although these will be considered a strictly harder problem and thus a secondary challenge.

Figure 3.1: Appearance variation as a result of changing the scene, rigid pose, within-class identity and non-rigid pose.

3.2 Sliding-Window Classification

The most straightforward technique to identify the regions of an image which contain an object class is simply to test every region in a densely sampled set. Thus the problem of detection is formulated as binary classification, where the positive class is the object and the negative class is anything else. Sliding-window classification refers to the particular case where the regions are rectangles of a fixed size at displacements on a regular grid.

Binary classification is often posed as learning a score function that maps inputs to a real number representing the confidence that it belongs to the positive class. This score can be thresholded to obtain the final classifications, although a classifier is often evaluated in terms of the precision-recall curve that is traced as the threshold is varied.

One pitfall of the sliding-window approach is that the necessary density of the windows that are tested causes a single instance to elicit multiple detections, due to the similarity of adjacent windows. This is avoided using a procedure known as Non-Maxima Suppression (NMS), which seeks to limit potential detections to the local maxima of the score function. NMS tries to ensure that the same pixels cannot belong to multiple instances, or at least that not too many of an instance's pixels can be shared.

If the scoring function depends linearly on the image pixels $f(x) = w^T x$, then evaluating the score of every window of a fixed size at one-pixel intervals amounts to convolution, for which efficient routines exist. A linear function for classification may be learnt using, for example, a linear Support Vector Machine (SVM), least-squares regression or logistic regression.

When the distance of an object from the camera is large compared to its size, the action of perspective projection is simply to scale its appearance inversely proportional to the distance. This is accounted for using multi-scale search, where sliding-window classification is performed on several resized versions of each image. NMS is applied to the scores from all scales together.

Unfortunately, sliding-window classification is not well suited to objects whose bounding box may vary in its aspect ratio. While it is possible to apply the detector to stretched images, this expands the domain on which the classifier is expected to function. It is also undesirable computationally because it adds an extra dimension that must be exhaustively searched at test time. Often a change in aspect ratio indicates a change in the rigid pose of an oblong object such as a bicycle or a pencil. Therefore a standard solution is to divide the examples into two or three groups based on their aspect ratio, train a detector for each and combine their results.

3.3 Efficient Non-Linear Classification

Much research effort has been devoted to the design of feature transforms for images due to the fact that linear functions of the raw pixel intensities are simply too restrictive to achieve good performance. It is therefore necessary to consider ways to learn a function that depends non-linearly on its input image. Due to the large number of windows which must be scored in object

detection, it is important to maintain efficiency of function evaluation.

A simple way to make the score function non-linear is to introduce an explicit feature map ϕ and then learn a linear function $f(x) = w^T\phi(x)$. Of particular interest are feature maps Φ that transform the original image x into a multi-channel feature image whose every pixel is obtained as the same function evaluated on the shifted input image

$$\Phi(x)[u] = \psi(L_u x) \tag{3.1}$$

with the linear translation operator defined $L_\tau x[u] = x[u + \tau]$. This may be considered non-linear convolution, since if $\psi(x) = a^T x$ then $\Phi(x) = a \star x$. It is common for ψ to have compact support. Feature maps of this form commute with translation

$$L_\tau \Phi(x) = \Phi(L_\tau x) \tag{3.2}$$

and therefore sliding-window evaluation of the score function is still convolution

$$w^T \Phi(L_\tau x) = w^T L_\tau \Phi(x) = \sum_{p=1}^{k} (w_p \star \Phi_p(x))[\tau] \tag{3.3}$$

where $\Phi_p(x)$ represents channel p of the k-channel feature image $\Phi(x)$. Not only does this enable the use of efficient routines to compute the convolution itself, it also allows overlapping windows to share the computational burden of the feature transform. The reason that these feature maps are of particular interest is that they allow training algorithms based on convolution and stationarity to extend beyond linear functions.

An alternative feature map that lends itself to efficient evaluation is one that depends on the image through sums of its pixels in rectangular regions. These sums can be obtained in constant time using summed area tables, also known as integral images. However, the rectangular sums are merely a linear

transformation of the input $\phi(x) = Ax$, and therefore it would make little sense to combine these with a linear classifier $w^T Ax = (A^T w)^T x$. These features are instead often used with (ensembles of) decision trees [18] and applied to feature images not the original images. A special case of this type of feature map is a global histogram per channel. Features of this nature lend themselves to Efficient Subwindow Search [37], which uses Branch and Bound to go beyond sliding-window and consider all rectangles. While efficient to evaluate, these feature maps are incompatible with training algorithms based on convolution and stationarity.

Kernels methods are another alternative for learning non-linear score functions, although these may be computationally prohibitive for detection due to the need to evaluate the kernel function with every support vector. There do exist methods to find an approximate explicit feature map for standard kernels such as the Gaussian, Laplacian, histogram intersection and Hellinger kernels [51, 66]. Kernels and kernel approximations that depend on the image through inner products could be evaluated in sliding-window fashion using convolution. However, they may still require a large number of convolutions compared to an explicit feature map. Generic kernels tend to be ineffective for images anyway due to their high dimensionality: the curse of dimensionality implies that the number of required examples grows exponentially with the dimension of the data.

3.4 Hard Negative Mining

Learning a linear classifier for detection is complicated by the open-ended negative set. In practice, this set is typically specified as every window in a set of negative images that are known not to contain the object. The

much smaller positive set is simply a collection of annotated windows that contain the object. The enormity of the negative set presents serious practical challenges.

Hard Negative Mining (HNM) is a heuristic which seeks to replace the full negative set with a smaller representative set. Each iteration of HNM uses the current detector to search the negative set for the examples with the worst (highest) scores. These are added to the training set and the next detector is trained. The underlying assumption is that evaluating the classifier on every example several times will be much faster than training with every example. The initial detector is learned using a random subset of negative windows.

The training algorithm within HNM is almost exclusively an SVM, due to its possible interpretation as identifying support vectors. However, it can also be understood as a way of incorporating NMS into the training procedure. This is important because NMS effectively alters the distribution that is encountered at testing time.

While it is possible to enforce uniqueness (i.e. of the position of windows, not of their contents) when adding negative examples to the active set, this could not prevent the inclusion of two adjacent windows that overlap almost entirely and are nearly identical in appearance. One solution would be to add the new negatives to the set, re-score all negatives using the current detector, and then perform NMS. However, this is likely to simply discard most of the old hard negatives, since the new hard negatives were *obtained* using the current detector and NMS. Therefore, uniqueness will not be enforced when adding hard negatives to the active set. This violates somewhat the interpretation that HNM identifies the support vectors.

There are many decisions which must be made in the design of an HNM

pipeline:

- what is the relative weight of positive and negative examples?
- what is the weight of the regularisation term?
- how many initial negatives should be used?
- how many hard negatives should be collected in each round?
- should hard negatives be kept from all rounds or only the previous round?
- should only negative examples with a minimum score (e.g. -1, 0 or 1) be considered hard?
- should the initial negatives be penalised in a separate term, or should they be treated as hard negatives from a previous round?

If the initial negatives are preserved in a separate term, then

- what is the relative weight of the initial and hard negatives?

If negative examples are amassed over multiple rounds, then the classification loss must be normalised to ensure that the cost of the negative examples does not dwarf the other terms over time.

The loss function which is minimised in each iteration of HNM is

$$\gamma_1 E(w, \mathcal{Z}_{\text{pos}}) + (1-\gamma_1)\left[\gamma_2 E(w, \mathcal{Z}_{\text{init}}) + (1-\gamma_2) E(w, \mathcal{Z}_{\text{hard}})\right] + \tfrac{1}{2}\lambda\|w\|^2 \quad (3.4)$$

with the classification loss function defined

$$E(w, \mathcal{Z}) = \frac{1}{|\mathcal{Z}|} \sum_{(x,y)\in\mathcal{Z}} \max(0, 1 - y w^T \phi(x)) \ . \quad (3.5)$$

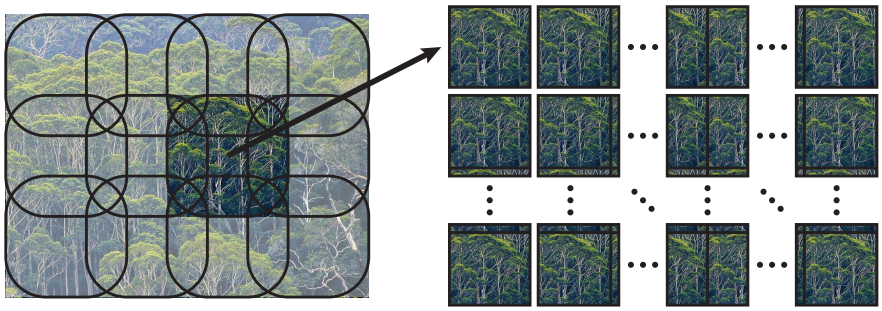


Figure 3.2: Henriques et al. approximated a densely-sampled set of windows in an image with all circular shifts (right) of a coarsely-sampled set of windows which cover the image (left). Rounded rectangles illustrate overlap.

3.5 Correlation Filters

Correlation Filters [40, 41] are a family of methods for learning a function from a set of images. Their defining trait is that they take advantage of the circulant structure which arises when all circular shifts of all example images are incorporated into the training set.

Henriques et al. [31] suggested that Correlation Filters could be used to efficiently train a sliding-window detector that makes use of the full set of negative examples. They proposed to approximate the dense set of sliding-window examples with the circular shifts of a coarse set, as depicted in Figure 3.2. Whereas HNM is a heuristic algorithm without a clear objective and may take several passes of the negative set to reach its optimal solution, Correlation Filters only require a single pass to obtain their global optimum. Unfortunately, Correlation Filters demand that the hinge loss be replaced by a least-squares loss, which is not as well suited to classification.

3.5.1 Least-Squares Regression

The simplest instance of Correlation Filters is in least-squares regression. The formulation of least-squares regression will be briefly developed before incorporating circular shifts. The goal is to learn an affine function $f(x) = w^T x + b$ given a training set of n input vectors $x_i \in \mathbb{R}^m$ and corresponding output labels $y_i \in \mathbb{R}$. Adopting a squared loss function, the template w and bias b are chosen to minimise the empirical loss

$$\frac{1}{2}\sum_{i=1}^{n}(x_i^T w + b - y_i)^2 \ . \tag{3.6}$$

It will sometimes be convenient to adopt the vector notation

$$\frac{1}{2}\|X^T w + b1 - y\|^2 \tag{3.7}$$

where X is an $m \times n$ matrix with the inputs as columns, y is a vector of all labels and 1 is a vector of ones. Setting to zero the derivative with respect to b yields

$$(X^T w + b1 - y)^T 1 = 0 \quad \Rightarrow \quad b = \frac{1}{n}\left(y^T 1 - w^T X 1\right) = \bar{y} - w^T \bar{x} \tag{3.8}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean input and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean label. Using this analytic expression to eliminate b reveals that learning an affine function $f(x) = w^T x + b$ produces the same template w as learning a linear function $f(x) = w^T x$ with the centroids removed from the inputs and labels, giving the empirical loss

$$\frac{1}{2}\sum_{i=1}^{n}[(x_i - \bar{x})^T w - (y_i - \bar{y})]^2 \ . \tag{3.9}$$

Ridge regression introduces regularisation to ensure stability of the optimisation procedure and reduce the variance of the generalisation error

$$\frac{1}{2n}\sum_{i=1}^{n}[(x_i - \bar{x})^T w - (y_i - \bar{y})]^2 + \frac{\lambda}{2}\|w\|^2 \ . \tag{3.10}$$

To solve for the template w involves the unconstrained minimisation of a convex quadratic function

$$\arg\min_{w} \quad \frac{1}{2} w^T (S + \lambda I) w - w^T r + \text{const.} \tag{3.11}$$

whose minimiser satisfies the linear system of equations $(S+\lambda I)w = r$ where

$$S = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \tag{3.12}$$

is the covariance of all examples, regardless of their label, and

$$r = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) \tag{3.13}$$

is a weighted combination of the centred examples.

Let $P = I - \frac{1}{n} 1 1^T$ denote the symmetric $n \times n$ projection operator that removes the centroid. This operator satisfies $P^2 = P$ and $P1 = 0$. Returning to vector notation, the un-normalised empirical loss is

$$\frac{1}{2} \| (X - \bar{x} 1^T)^T w - (y - \bar{y} 1) \|^2 = \frac{1}{2} \| X^T w - y \|_P^2 \ . \tag{3.14}$$

Under this notation $S = \frac{1}{n} X P X^T$ and $r = \frac{1}{n} X P y$. It becomes clear that in the expression for the right-hand side r, it is sufficient to subtract the centroid from either the inputs or the outputs

$$r = \tfrac{1}{n}(XP)y = \frac{1}{n} \sum_{i=1}^{n} y_i (x_i - \bar{x}) = \tfrac{1}{n} X(Py) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y}) x_i \ . \tag{3.15}$$

It is also evident that the means can be subtracted after the summation in both expressions

$$r = \tfrac{1}{n} X (I - \tfrac{1}{n} 1 1^T) y \quad = \frac{1}{n} \sum_{i=1}^{n} y_i x_i - \bar{y}\bar{x} \tag{3.16}$$

$$S = \tfrac{1}{n} X (I - \tfrac{1}{n} 1 1^T) X^T = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T - \bar{x}\bar{x}^T \ , \tag{3.17}$$

which is useful for constructing the system of equations in a single pass through the data.

3.5.2 Least-Squares Correlation Filter

The least-squares Correlation Filter is a special case of least-squares regression where the inputs are images and the training set is derived from the circular shifts of a set of base examples. Let $x_i : \mathcal{U} \to \mathbb{R}$ for $i = 1, \ldots, n$ denote the base examples, which are all of size $\ell = (\ell_1, \ell_2)$ with domain $\mathcal{U} = \{u \in \mathbb{Z}^2 : 0 \leq u < \ell\}$. Define the periodic translation operator $L_\tau x[u] = x[u + \tau \bmod \ell]$. Each base example has a label per shift $y_i : \mathcal{U} \to \mathbb{R}$ such that $y_i[\tau]$ is the label for input $L_\tau x_i$. A typical label assignment for classification tasks is described in Figure 3.3. The loss per base example x_i can be expressed in terms of circular cross-correlation

$$\sum_{\tau \in \mathcal{U}} ((L_\tau x_i)^T w - y_i[\tau])^2 = \|w \star x_i - y_i\|^2 \ . \tag{3.18}$$

Using the results of the previous section, a linear function $f(x) = w^T x$ will be studied instead of an affine function, with subsequent consideration given to the removal of the centroid. Let X_i be the $m \times m$ two-level circulant Hankel matrix such that $X_i w = w \star x_i$ as defined in Section 2.13, where $m = \ell_1 \ell_2 = |\mathcal{U}|$ is the number of pixels in each image. Then the overall problem can be expressed

$$\arg\min_w \quad \frac{1}{2mn} \sum_{i=1}^{n} \|X_i w - y_i\|^2 + \frac{\lambda}{2} \|w\|^2 \tag{3.19}$$

and its unique solution satisfies $(S + \lambda I)w = r$ where

$$S = \frac{1}{mn} \sum_{i=1}^{n} X_i^T X_i \ , \qquad r = \frac{1}{mn} \sum_{i=1}^{n} X_i y_i \ . \tag{3.20}$$

The symmetric matrix S is (two-level) circulant Toeplitz matrix with elements $S[u, t] = s[u - t \bmod \ell]$ because the product of two Hankel operators is a Toeplitz operator (Section 2.13). Both s and r are obtained via circular

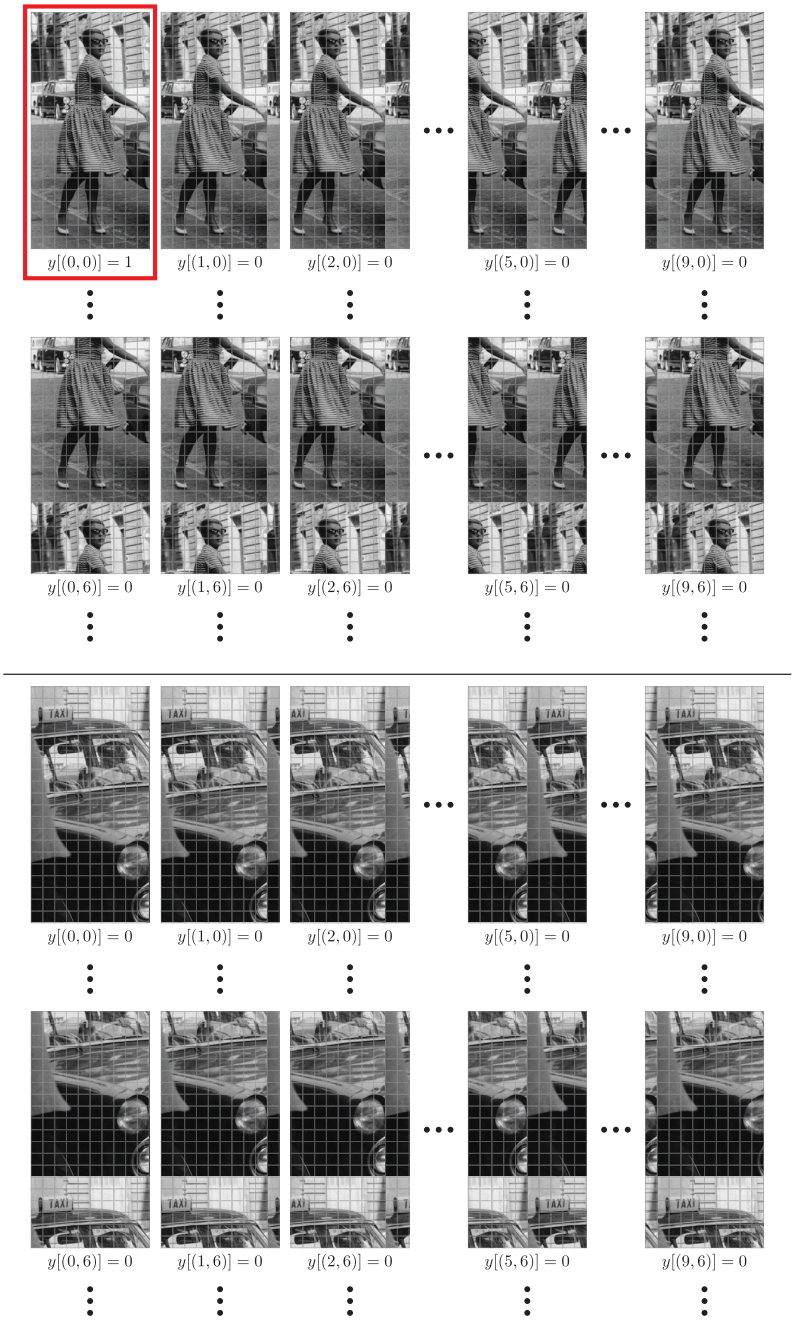


Figure 3.3: Correlation Filters are trained on all circular shifts of a set of base examples of uniform size, with every shift assigned a label. For a positive base example (top), the unshifted example belongs to the positive class and all other shifts to the negative class. For a negative base example (bottom), every shift belongs to the negative class.

cross-correlation

$$s = \frac{1}{mn} \sum_{i=1}^{n} x_i \star x_i \ , \qquad r = \frac{1}{mn} \sum_{i=1}^{n} y_i \star x_i \ . \tag{3.21}$$

Therefore $Sw = s * w$ and the system of equations is diagonalised by the DFT

$$(\mathrm{diag}(\hat{s}) + \lambda I)\hat{w} = \hat{r} \ . \tag{3.22}$$

Whereas algorithms to solve general $m \times m$ systems take $O(m^3)$ time, this diagonal system can be solved in $O(m)$ time using element-wise division

$$\hat{w}[u] = \frac{\hat{r}[u]}{\hat{s}[u] + \lambda} \quad u \in \mathcal{U} \ , \tag{3.23}$$

with $O(m \log m)$ time required for forward and inverse transforms. Since both $\hat{s}$ and $\hat{r}$ can be obtained efficiently from the Fourier transforms of the examples

$$\hat{s} = \frac{1}{mn} \sum_{i=1}^{n} \hat{x}_i^* \odot \hat{x}_i \ , \qquad \hat{r} = \frac{1}{mn} \sum_{i=1}^{n} \hat{y}_i^* \odot \hat{x}_i \ , \tag{3.24}$$

the entire algorithm can operate in the Fourier domain. The final template $w = F^{-1}\hat{w}$ is real if and only if the solution $\hat{w}$ in (3.23) has conjugate symmetry. This is guaranteed since $\hat{r}$ has conjugate symmetry and $\hat{s}$ is real and symmetric. In fact, the complex linear system of equations can be transformed into an equivalent real system of the same size by separating real and imaginary components and exploiting conjugate symmetry.

Centroid Removal

The loss function considering an affine instead of linear function is

$$\sum_{i=1}^{n} \| w \star x_i + b1 - y_i \|^2 \ . \tag{3.25}$$

The expression for the bias term becomes $b = \bar{y} - \bar{x}1^T w$, where $\bar{x} \in \mathbb{R}$ is the mean *pixel* rather than the mean *example*, and $\bar{y} \in \mathbb{R}$ is similarly the mean label

$$\bar{x} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{u \in \mathcal{U}} x_i[u] \ , \qquad \bar{y} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{u \in \mathcal{U}} y_i[u] \ . \tag{3.26}$$

Making this substitution, the loss is

$$\sum_{i=1}^{n} \| w \star (x_i - \bar{x}1) - (y_i - \bar{y}1) \|^2 \ . \tag{3.27}$$

This can be understood as removing the average zero-frequency (DC) component from both images and labels. From the regular least-squares case, recall that it is sufficient to centre just the images or the labels. The centroid removal can also be performed after the fact using

$$s = \frac{1}{mn} \sum_{i=1}^{n} (x_i - \bar{x}1) \star (x_i - \bar{x}1) = \frac{1}{mn} \sum_{i=1}^{n} x_i \star x_i - \bar{x}^2 1 \ , \tag{3.28}$$

$$r = \frac{1}{mn} \sum_{i=1}^{n} (y_i - \bar{y}1) \star (x_i - \bar{x}1) = \frac{1}{mn} \sum_{i=1}^{n} y_i \star x_i - \bar{y}\bar{x}1 \ . \tag{3.29}$$

In the Fourier domain this only affects the zero-frequency (DC) component since $F1 = m\delta$, where δ is the Dirac delta. Using $\hat{\bar{x}}[0] = m\bar{x} \in \mathbb{R}$ to denote the mean zero-frequency component, where $\hat{\bar{x}} = \frac{1}{n} \sum_i (F x_i) = F(\frac{1}{n} \sum_i x_i)$ is both the mean of the Fourier transforms and the Fourier transform of the mean, gives

$$\hat{s}[u] = \begin{cases} \dfrac{1}{mn} \displaystyle\sum_{i=1}^{n} \hat{x}_i^*[0] \hat{x}_i[0] - \frac{1}{m} \hat{\bar{x}}[0]^2, & u = 0 \\ \dfrac{1}{mn} \displaystyle\sum_{i=1}^{n} \hat{x}_i^*[u] \hat{x}_i[u], & u \neq 0 \end{cases} \tag{3.30}$$

and similarly

$$\hat{r}[u] = \begin{cases} \dfrac{1}{mn} \displaystyle\sum_{i=1}^{n} \hat{y}_i^*[0] \hat{x}_i[0] - \frac{1}{m} \hat{\bar{y}}[0] \hat{\bar{x}}[0], & u = 0 \\ \dfrac{1}{mn} \displaystyle\sum_{i=1}^{n} \hat{y}_i^*[u] \hat{x}_i[u], & u \neq 0 \ . \end{cases} \tag{3.31}$$

Henriques et al. [31] simply set $\hat{y}_i[0] = 0$ for every example i instead of subtracting the mean.

3.5.3 Multi-Channel Correlation Filters

The recent extension to Multi-Channel Correlation Filters [31, 34, 5] crucially enables the technique to be applied to feature images with more than one channel. Let each input $x_i : \mathcal{U} \to \mathbb{R}^k$ now be a k-channel signal with elements $x_{ip}[u] \in \mathbb{R}$. The template $w : \mathcal{U} \to \mathbb{R}^k$ is a signal of the same class. Channel p of x_i is denoted $x_{ip} : \mathcal{U} \to \mathbb{R}$. The Multi-Channel Correlation Filter objective for each base example i is

$$\left\| \sum_{p=1}^{k} w_p \star x_{ip} - y_i \right\|^2 = \left\| \sum_{p=1}^{k} X_{ip} w_p - y_i \right\|^2 \tag{3.32}$$

where X_{ip} is the two-level circulant Hankel matrix $X_{ip} = X_{ip}^T$ such that $X_{ip} w_p = w_p \star x_{ip}$. The overall regularised objective is

$$\frac{1}{2mn} \sum_{i=1}^{n} \left(\sum_{p=1}^{k} \sum_{q=1}^{k} w_p^T X_{ip} X_{iq} w_q - 2 y_i^T \sum_{p=1}^{k} X_{ip} w_p \right) + \frac{\lambda}{2} \|w\|^2 \tag{3.33}$$

which, re-ordering summations, can be expressed

$$\frac{1}{2} \sum_{p=1}^{k} \sum_{q=1}^{k} w_p^T S_{pq} w_q - \sum_{p=1}^{k} w_p^T r_p + \frac{\lambda}{2} \|w\|^2 \tag{3.34}$$

introducing

$$S_{pq} = \frac{1}{mn} \sum_{i=1}^{n} X_{ip} X_{iq} \ , \qquad r_p = \frac{1}{mn} \sum_{i=1}^{n} X_{ip} y_i \ . \tag{3.35}$$

Each S_{pq} is circulant Toeplitz with elements $S_{pq}[u, t] = s_{pq}[u - t \bmod \ell]$ due to the circulant Hankel structure of each X_{ip}. The objective is therefore simply $\frac{1}{2} w^T (S + \lambda I) w - r^T w$ where S is *block multi-level circulant*. The elements of

s_{pq} and r_p are obtained by circular cross-correlation

$$s_{pq} = \frac{1}{mn} \sum_{i=1}^{n} x_{iq} \star x_{ip} \ , \qquad \hat{s}_{pq} = \frac{1}{mn} \sum_{i=1}^{n} \hat{x}_{iq}^{*} \odot \hat{x}_{ip} \ , \tag{3.36}$$

$$r_p = \frac{1}{mn} \sum_{i=1}^{n} y_i \star x_{ip} \ , \qquad \hat{r}_p = \frac{1}{mn} \sum_{i=1}^{n} \hat{y}_i^{*} \odot \hat{x}_{ip} \ , \tag{3.37}$$

analogous to (3.21) and (3.24). The symmetry of S implies that $S_{pq} = S_{qp}^{T}$, $s_{pq}[u] = s_{qp}[-u \bmod \ell]$ and $\hat{s}_{pq} = \hat{s}_{qp}^{*}$.

The unique minimiser of this quadratic objective satisfies $(S + \lambda I)w = r$, or

$$\sum_{q=1}^{k} S_{pq} w_q + \lambda w_p = r_p \qquad p = 1, \ldots, k \ . \tag{3.38}$$

Using $F(S_{pq} w_q) = \hat{s}_{pq} \odot \hat{w}_q$, the equivalent problem in the Fourier domain is

$$\sum_{q=1}^{k} \hat{s}_{pq}[u] \, \hat{w}_q[u] + \lambda \hat{w}_p[u] = \hat{r}_p[u] \qquad u \in \mathcal{U}, \quad p = 1, \ldots, k \tag{3.39}$$

from which it is apparent that each $\hat{w}[u]$ can be solved independently

$$\hat{w}[u] = (\hat{s}[u] + \lambda I)^{-1} \hat{r}[u] \qquad u \in \mathcal{U} \tag{3.40}$$

in a direct generalisation of (3.23) from scalar division ($k = 1$) to $k \times k$ matrix inversion. Here $\hat{s}[u]$ is a block of the multi-level circulant matrix $\hat{S}$ with elements $\hat{S}_{pq}[u, t] = \hat{s}_{pq}[u - t \bmod \ell]$. This can be understood as block-diagonalising the $mk \times mk$ problem to yield m independent problems of size $k \times k$ [9]. The final solution is obtained by taking the inverse transform $w_p = F^{-1} \hat{w}_p$ for each channel p, which is again guaranteed to be real.

Subtraction of the centroid to learn affine instead of linear Multi-Channel Correlation Filters is deferred to Appendix A.1.

3.5.4 Complexity Analysis

To construct the system, all channels of each example must be transformed, taking $O(nkm \log m)$ time, and the channel pairs must be multiplied to form

$\hat{s}_{pq}$, taking $O(nk^2m)$ time, giving a total of $O(nkm(k + \log m))$ time. To solve the system takes $O(k^3m)$ time and to compute the inverse transforms takes $O(km \log m)$ time, giving a total of $O(km(k^2 + \log m))$ time to obtain a solution. If the number of examples is much larger than the number of channels $n \gg k$, then the time to build the system dominates the time to solve it.

To train a Correlation Filter for object detection, it is necessary to sample windows from a set of negative images. If the windows have m pixels and are sampled to cover images with M pixels, then there will be $n \in O(M/m)$ examples per negative image. Therefore to construct the linear system for a Correlation Filter with k channels will take $O(kM(k + \log m))$ time per negative image.

3.6 Stationary Process LDA

Hariharan et al. [29] proposed another efficient alternative to HNM for learning a detector. They considered sliding-window classification in the simple generative framework of Linear Discriminant Analysis (LDA). Their key contribution was to assume that the distribution encountered by a sliding-window classifier is translation-invariant, or equivalently that the random variable defined by sampling a random window in a random image is a stationary process. Figure 3.4 presents the intuition behind this assumption. This section reviews the formulation of their method, which is dubbed Stationary Process LDA (SPLDA).

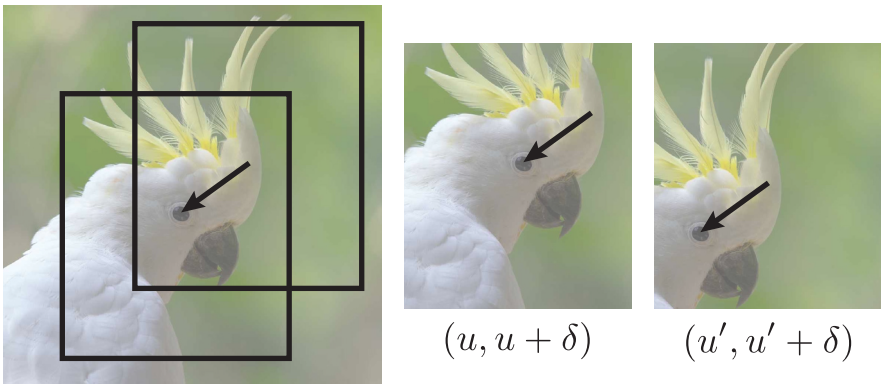


Figure 3.4: The covariance of the distribution of images encountered by a sliding-window classifier is roughly translation invariant. The same evidence that is observed at pixels $x[u]$ and $x[u+\delta]$ in one window will be observed at pixels $x[u']$ and $x[u'+\delta]$ in another window.

3.6.1 Linear Discriminant Analysis

LDA is a fundamental generative approach to binary classification. It begins with the assumption that the conditional distribution of each class $j \in \{1, 2\}$ is multivariate Gaussian

$$p_{X|Y}(x|j) = \mathcal{N}(x; \mu_j, \Sigma) = \frac{1}{Z(\Sigma)} \exp\left[-\frac{1}{2}(x-\mu_j)^T \Sigma^{-1}(x-\mu_j)\right] \quad (3.41)$$

with mean μ_j and covariance Σ. The denominator $Z(\Sigma)$ normalises the function to have unit integral. Having a shared covariance Σ ensures that the discriminant is affine. The marginal distribution of the classes is controlled with a single parameter

$$p_Y(j) = \begin{cases} \alpha, & j = 1, \\ 1-\alpha, & j = 2 \ . \end{cases} \quad (3.42)$$

Thus the model is fully specified in the parameters $\theta = (\mu_1, \mu_2, \Sigma, \alpha)$. The discriminant $p_{Y|X}(1|x) > p_{Y|X}(2|x)$ is formulated using Bayes' rule

$$p_{X|Y}(x|1)\, p_Y(1) > p_{X|Y}(x|2)\, p_Y(2) \tag{3.43}$$

and considering the log likelihoods

$$-\tfrac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)+\ln\alpha > -\tfrac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)+\ln(1-\alpha) \tag{3.44}$$

which, with some manipulation, yields the affine inequality

$$x^T\Sigma^{-1}(\mu_1-\mu_2) > \tfrac{1}{2}(\mu_1+\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) + \ln[(1-\alpha)/\alpha] \ . \tag{3.45}$$

Therefore the discriminant takes the form $f(x) = w^T x + b$ and the model predicts $j = 1$ if $f(x)$ is positive and $j = 2$ if $f(x)$ is negative. The most important parameter of $f(x)$ is the template

$$w = \Sigma^{-1}(\mu_1 - \mu_2) \tag{3.46}$$

or to be precise, the direction of the template $w/\|w\|$. The bias b and magnitude $\|w\|$ can be disregarded due to the evaluation of classifiers in terms of precision-recall curves. Inner products with the template can be interpreted as projections on to the line connecting the two class means, measured in a whitened space where the Gaussian distributions have isotropic covariance

$$x^T w = (\Sigma^{-\frac{1}{2}}x)^T(\Sigma^{-\frac{1}{2}}(\mu_1-\mu_2)) \ . \tag{3.47}$$

Given a training set of n examples (x_i, y_i), the parameters of the model are typically chosen to maximise the joint likelihood of the evidence $p_{X,Y}(x, y; \theta)$ assuming independence of the examples

$$\arg\max_\theta \prod_{i=1}^{n} p_{X|Y}(x_i|y_i;\theta)\, p_Y(y_i;\theta) \ . \tag{3.48}$$

Equivalently minimising the negative log likelihood, it becomes apparent that the problems for (μ_1, μ_2, Σ) and α are separable

$$\begin{aligned}&\min_{\mu_1,\mu_2,\Sigma,\alpha} \sum_{i=1}^{n} \Big\{ -\ln p_{X|Y}(x_i|y_i;\theta) - \ln p_Y(y_i;\theta) \Big\} \\ &= \min_{\mu_1,\mu_2,\Sigma} \sum_{j=1,2} \sum_{i\in\mathcal{C}_j} -\ln \mathcal{N}(x_i;\mu_j,\Sigma) + \min_{\alpha} \left\{ -n_1 \ln \alpha - n_2 \ln(1-\alpha) \right\} \quad .\end{aligned} \tag{3.49}$$

Here $\mathcal{C}_j = \{i : y_i = j\}$ is the set of examples in class j and $n_j = |\mathcal{C}_j|$ is the number. For any choice of α, the optimal mean of each conditional distribution is its empirical mean $\bar{x}_j$, and the optimal shared covariance is the empirical "within-class" covariance

$$S_W = \frac{1}{n} \sum_{j=1,2} \sum_{i\in\mathcal{C}_j} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^T = \frac{1}{n} \left(n_1 S_1 + n_2 S_2 \right) \quad . \tag{3.50}$$

where S_j is the empirical covariance of each class. The optimal $\alpha = n_1/n$, however this affects only the threshold of the decision function and so can be ignored. Hence the template would be computed from data

$$w = S_W^T (\bar{x}_1 - \bar{x}_2) \quad . \tag{3.51}$$

An apparent difference between LDA and least-squares regression is that LDA uses the within-class covariance matrix in (3.50), which depends on the labels, whereas least-squares regression uses the unsupervised covariance of all examples in (3.12). However, an historical result [20] states that the LDA solutions obtained using either covariance matrix are equivalent

$$(S + \lambda I)^{-1} (\bar{x}_1 - \bar{x}_2) \propto (S_W + \lambda I)^{-1} (\bar{x}_1 - \bar{x}_2) \quad . \tag{3.52}$$

See Appendix A.2 for a proof.

Furthermore, if the labels in least-squares regression take only two values $y_i \in \{\gamma_1, \gamma_2\}$ corresponding to two classes with means $\bar{x}_1$ and $\bar{x}_2$, then the

right-hand side is equivalent to that of LDA

$$r \propto \bar{x}_1 - \bar{x}_2 \ . \tag{3.53}$$

See Appendix A.3 for this proof. This right-hand side can also be computed using the mean of all examples $r = \bar{x}_1 - \bar{x}$ without affecting the classifier, since

$$\bar{x}_1 - \bar{x} = \bar{x}_1 - \frac{n_1\bar{x}_1 + n_2\bar{x}_2}{n_1 + n_2} = \frac{n_2}{n_1 + n_2}(\bar{x}_1 - \bar{x}_2) \ . \tag{3.54}$$

3.6.2 Stationarity

Hariharan et al. [29] recognised that the distribution of negative examples is stationary. Therefore its mean image is uniform (takes the same value at all pixels) and its covariance matrix is a block multi-level Toeplitz matrix. Furthermore, the distribution of *all* examples is stationary, since every positive example belongs to a sliding-window set in the image from which it came, and therefore the overall covariance matrix S has the same structure. This has an alternative interpretation as the large negative set dominating estimation of the within-class covariance matrix.

The first advantage of a Toeplitz covariance matrix with elements $S_{pq}[u,t] = s_{pq}[u-t]$ is that it has far fewer parameters to estimate and store. Consider k-channel images with $m = \ell_1\ell_2$ pixels. Whereas a full covariance matrix has $O(k^2m^2)$ elements, its Toeplitz counterpart is defined by the $O(k^2m)$ elements of s.

Besides the diminished number of parameters, a distinct advantage of the Toeplitz structure is that the covariance for a window of any size is trivially obtained as a sub-matrix of the covariance for a larger window. This is due to the well-known result that if X is a random vector distributed according to $\mathcal{N}(\mu, \Sigma)$ and P is a matrix of appropriate dimension, then PX is distributed

according to $\mathcal{N}(P\mu, P\Sigma P^T)$. If the matrix P selects a subset of elements from a vector, then $P\Sigma P^T$ selects a sub-matrix of Σ. This means that the covariance of all examples only needs to be estimated once *ever*, for the largest conceivable window size.

3.6.3 Estimation of Toeplitz Covariance

An empirical Toeplitz covariance matrix can be estimated from a set of images $x_1, \ldots, x_n$ of arbitrary size by computing the covariance $s[\delta]$ of all pixel pairs $(x_i[u], x_i[u+\delta])$ with relative displacement δ. These are assembled to form a "patchwork" Toeplitz covariance matrix. The covariance of each relative displacement δ is estimated from n images according to

$$s[\delta] = \left(\sum_{i=1}^{n} \sum_{u, u+\delta \in \mathcal{D}_i} x_i[u]\, x_i[u+\delta]^T \right) \Bigg/ \left(\sum_{i=1}^{n} \sum_{u, u+\delta \in \mathcal{D}_i} 1 \right) - \bar{x}\bar{x}^T \qquad (3.55)$$

where $\bar{x} \in \mathbb{R}^k$ is the mean pixel. This simultaneously regularises the estimate of the covariance matrix and maximises the amount of evidence for each parameter. If the domain of the image is $\mathcal{U} = \{u \in \mathbb{Z}^2 : 0 \leq u < \ell\}$, then the covariance must be estimated for $\delta \in \mathcal{U} - \mathcal{U} = \{\delta \in \mathbb{Z}^2 : |\delta| < \ell\}$.

Estimating the individual covariance for each relative displacement δ using a different number of observations is analogous to covariance estimation from vectors with missing data. Pixels beyond the edge of the image are effectively "missing" and do not contribute to the numerator or denominator. However, this method of covariance estimation with missing data is not guaranteed to yield a matrix which is positive semidefinite.

Chapter 4

Comparative Study of Toeplitz Covariance Methods

4.1 Introduction

There are stark similarities between Correlation Filters and SPLDA. This chapter will undertake a thorough comparative study and develop connections between them. Both methods compute their template $w = (S + \lambda I)^{-1} r$ where r is a linear combination of examples and S is a Toeplitz covariance matrix. Both algorithms consist of two main phases: constructing the linear system and solving it. However, there is a fundamental difference in how the two methods obtain their covariance. Whereas SPLDA uses images of arbitrary size and makes no assumption about what lies beyond the boundary of an image, Correlation Filters use images of the same size as the positive examples and assume periodic extension. This distinction has several practical ramifications.

The first critical difference is that Correlation Filters receive a significant computational advantage from the assumption of periodic signals. This

assumption manifests in a covariance matrix which is not just Toeplitz but *circulant* Toeplitz. Since every circulant matrix has the DFT basis as its eigenvectors, the template can be computed efficiently via the FFT. In contrast, the template in SPLDA is computed by explicitly instantiating the large matrix and performing Cholesky factorisation, which demands much more computational effort and memory. Furthermore, estimating the covariance matrix in Correlation Filters amounts to circular cross-correlation, meaning that the system can also be constructed efficiently in the Fourier domain.

A second important distinction is that the Toeplitz covariance matrix in SPLDA is not specific to the size of the input image. The covariance for a smaller image size is a sub-matrix of the covariance for a larger image size. Therefore, once the Toeplitz covariance has been estimated for a large window, it never needs to be computed again. Existing Correlation Filter literature, on the other hand, does not suggest a method to avoid re-computing the covariance matrix to learn a detector with different dimensions. An additional elegant property of the (non-circulant) Toeplitz covariance is that it can be estimated using entire images of arbitrary size. On the other hand, to estimate the circulant covariance matrix from a set of large natural images, it is necessary to sample a set of windows of the desired size.

A third difference is simply that periodic extension of the examples will introduce spurious evidence to the estimation of the covariance. This seems likely to diminish the performance of the detector, although it's not obvious to what degree. When independently compared to Hard Negative Mining (HNM) for pedestrian detection on the INRIA dataset, SPLDA was reported to be slightly worse [29], and Correlation Filters were reported to be slightly better [31].

This chapter will establish several results which bridge the apparent divide between the two methods. First it will be shown that the FFT can also be used to efficiently estimate the (non-circulant) Toeplitz covariance matrix in SPLDA. Given such a matrix, it will then be demonstrated that it is trivial to obtain a *circulant* Toeplitz covariance matrix for any image size. This implies that it is not necessary to compute and store a circulant covariance matrix per image size in Correlation Filters. The problem of solving a Toeplitz system of equations will then be studied, and an efficient algorithm using Preconditioned Conjugate Gradient with an inverse circulant preconditioner will be presented. Finally, an empirical comparison which considers detection performance as well as time and memory will be conducted.

4.2 Efficient Estimation of the Toeplitz Covariance

Construction of the linear system in Correlation Filters is efficient due to the use of the FFT to compute the elements of the covariance matrix in the Fourier domain $\hat{s}_{pq}$. Unfortunately, the Fourier domain imposes periodic extension, and its use in Correlation Filters condemns the covariance matrix to depend on the size of the input images ℓ, having elements $S[u,t] = s[u-t \bmod \ell]$. In contrast, SPLDA estimates the covariance for relative displacements in the spatial domain $s[\delta]$ and its elements $S[u,t] = s[u-t]$ do not depend on the size of the input images. However, the Fourier domain can still be used to accelerate the estimation of this covariance matrix.

The expression for the estimation of the Toeplitz covariance matrix in (3.55) comprises a numerator and denominator

$$s_{pq}[\delta] = \left(\sum_{i=1}^{n} T_{ipq}[\delta]\right) \Bigg/ \left(\sum_{i=1}^{n} N_i[\delta]\right) - \bar{x}_p \bar{x}_q \ . \tag{4.1}$$

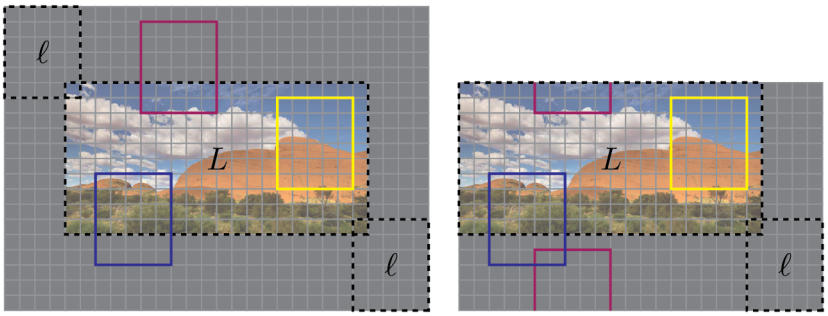


Figure 4.1: Every non-zero window of size ℓ in the infinite zero-padded extension of an image of size L (left) is contained once per period in the periodic extension of the image zero-padded to at least size $L+\ell-1$ (right).

Each term in the numerator can be formulated as infinite convolution

$$T_{ipq}[\delta] = \sum_{u,u+\delta\in\mathcal{D}_i} x_{ip}[u]\, x_{iq}[u+\delta] = (\tilde{x}_{ip} \star \tilde{x}_{iq})[\delta] \tag{4.2}$$

where $\tilde{x}_{ip}$ is the infinite extension of x_{ip} with zeros

$$\tilde{x}_{ip}[u] = \begin{cases} x_{ip}[u] & \text{if } u \in \mathcal{D}_i \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Recall that each image x_i has domain $\mathcal{D}_i \subset \mathbb{Z}^2$ of size $L_i = (L_{i1}, L_{i2}) \in \mathbb{Z}^2$.

This zero-padded infinite convolution can be computed as zero-padded periodic convolution. The image must be zero-padded to size $P_i \geq L_i + \ell - 1$. This is sufficient to ensure that $\tilde{x}_i[u+\delta \bmod P_i] = 0$ if $u+\delta \notin \mathcal{D}_i$ for any $u \in \mathcal{D}_i$ and $|\delta| < \ell$. This argument is made visually in Figure 4.1. An elegant feature of obtaining the covariance matrix by zero-padded periodic convolution is that the image can be padded to a larger size than strictly necessary. This is advantageous because practical FFT algorithms are significantly faster when the signal dimensions are composed of small prime factors such as 2, 3, 5 and 7.

Each term in the denominator can be determined in closed form as the number of elements in the set

$$\begin{aligned}\{u : u, u+\delta \in \mathcal{D}_i\} &= \{u : (0 \leq u < L_i) \wedge (-\delta \leq u < L_i - \delta)\} \\ &= \{u : \max(0, -\delta) \leq u < L_i + \min(0, -\delta)\} \ , \end{aligned} \tag{4.4}$$

which is simply

$$N_i[\delta] = \prod_{j=1}^{d} [L_{ij} - \max(0, \delta_j) - \max(0, -\delta_j)] = \prod_{j=1}^{d} (L_{ij} - |\delta_j|) \ . \tag{4.5}$$

Therefore to estimate the Toeplitz covariance matrix from a set of k-channel images, each with M pixels, takes $O(kM(k+\log M))$ time per image: $O(kM \log M)$ to compute transforms and $O(k^2 M)$ to perform element-wise multiplication of each channel pair. This compares well to the computational complexity of constructing the linear system for a Correlation Filter, which was shown to be $O(kM(k+\log m))$ for windows with m pixels in Section 3.5.4. Thus the non-circulant Toeplitz matrix can be estimated with similar effort to the circulant Toeplitz matrix, while remaining independent of the window size and avoiding the introduction of periodic boundary effects.

4.3 From Toeplitz to Circulant Toeplitz

The matrix in Correlation Filters is circulant because it is the covariance of all circular shifts of a set of images. This section formulates an expression for the covariance of all circular shifts of a set of images whose own covariance is specified in a Toeplitz matrix. In this way, the circulant covariance matrix for a Correlation Filter of any size can be constructed from a Toeplitz covariance matrix without having to sample an explicit set of windows.

Theorem 4.1. *If a set $\mathcal{X}$ of one-dimensional signals of length ℓ has Toeplitz covariance $G[u,t] = g[u-t]$, then the covariance of the set of all circular shifts of these signals $\{L_\tau x : x \in \mathcal{X}, \tau \in \mathcal{U}\}$ is circulant $H[u,t] = h[u - t \bmod \ell]$ with elements*

$$h[\delta] = (1-\theta)\, g[\delta] + \theta\, g[\delta - \ell] \tag{4.6}$$

with $\theta = \delta/\ell$ for $0 \le \delta < \ell$.

Proof. If a set $\mathcal{X}$ of n signals has covariance $G[u,t] = g[u-t]$, then

$$G[u,t] = \frac{1}{n}\sum_{i=1}^{n} x_i[u]\, x_i[t]^T = g[u-t] \ . \tag{4.7}$$

The covariance of the set of circular shifts is

$$\begin{aligned} H[u,t] &= \frac{1}{n\ell}\sum_{i=1}^{n}\sum_{\tau=0}^{\ell-1} x_i[\tau + u \bmod \ell]\, x_i[\tau + t \bmod \ell]^T \\ &= \frac{1}{n\ell}\sum_{i=1}^{n}\sum_{\tau=0}^{\ell-1} x_i[\tau + u - t \bmod \ell]\, x_i[\tau]^T \\ &= \frac{1}{\ell}\sum_{\tau=0}^{\ell-1} g[(\tau + u - t \bmod \ell) - \tau] \end{aligned} \tag{4.8}$$

This confirms that the covariance matrix is circulant $H[u,t] = h[u-t \bmod \ell]$ since $(\tau + \delta \bmod \ell) = (\tau + (\delta \bmod \ell) \bmod \ell)$. Observe that for $0 \le \tau < \ell$

$$(\tau + \delta \bmod \ell) - \tau = \begin{cases} \delta \bmod \ell & \text{if } (\tau + \delta \bmod \ell) \ge \tau \\ -(-\delta \bmod \ell) & \text{if } (\tau + \delta \bmod \ell) < \tau \ . \end{cases} \tag{4.9}$$

If, further, $0 \le \delta < \ell$, then

$$(\tau + \delta \bmod \ell) - \tau = \begin{cases} \delta & \text{if } \tau < \ell - \delta \\ \delta - \ell & \text{if } \tau \ge \ell - \delta \ . \end{cases} \tag{4.10}$$

since

$$(\tau + \delta \bmod \ell) \ge \tau \quad \Leftrightarrow \quad \tau < \ell - \delta \ . \tag{4.11}$$

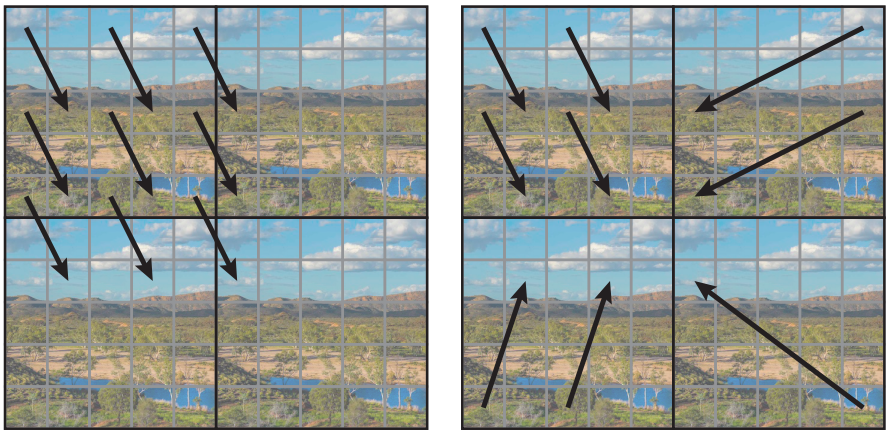


Figure 4.2: Under periodic extension, small apparent displacements (left) are sometimes observed as large actual displacements (right). Conversely, large apparent displacements are mostly observed as small actual displacements. The four relative displacements in the right image have the same covariance. The estimation of this covariance is biased towards the smaller actual displacements because they occur more often.

The elements of the circulant covariance matrix are therefore

$$h[\delta] = \frac{1}{\ell} \sum_{\tau=0}^{\ell-\delta-1} g[\delta] + \frac{1}{\ell} \sum_{\tau=\ell-\delta}^{\ell-1} g[\delta - \ell] \ . \tag{4.12}$$

□

The extension of this to two dimensions is

$$\begin{aligned} h[(\delta_1, \delta_2)] = \quad & (1-\theta_1)\,(1-\theta_2)\, g[(\quad \delta_1, \quad \delta_2)] \\ + \quad & \theta_1\,(1-\theta_2)\, g[(\delta_1 - \ell_1, \quad \delta_2)] \\ + \, & (1-\theta_1) \quad \theta_2\, g[(\quad \delta_1, \delta_2 - \ell_2)] \\ + \quad & \theta_1 \quad \theta_2\, g[(\delta_1 - \ell_1, \delta_2 - \ell_2)] \end{aligned} \tag{4.13}$$

with $\theta_j = \delta_j / \ell_j$. The intuition behind this convex combination is that, under periodic extension, pairs of pixels with relative displacement $0 \leq \delta < \ell$ will

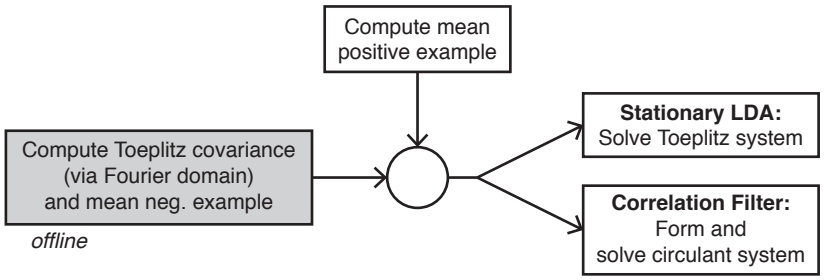


Figure 4.3: The unified pipeline for training a Correlation Filter or SPLDA classifier. Once the second-order statistics of natural images have been estimated offline, an explicit negative set is no longer required to train a detector.

have the same statistics as those with displacements $\delta-(\ell_1,0)$, $\delta-(0,\ell_2)$ and $\delta-(\ell_1,\ell_2)$ due to the element-wise modulo $H[u,t]=h[u-t \bmod \ell]$. Of these four displacements, smaller displacements will be observed more often and thus have a greater effect on the covariance. This is illustrated in Figure 4.2.

The expression in (4.13) enables the circulant covariance matrix for a Correlation Filter of arbitrary size to be obtained trivially from a Toeplitz covariance matrix, without having to choose and sample explicit negative examples. The right-hand side of the linear system in Correlation Filters is simply obtained $r=\bar{x}_1-\bar{x}$ where $\bar{x}_1$ is the mean of the positive examples and $\bar{x}$ is a uniform image whose every pixel is the mean pixel. This enables a unified pipeline for training Correlation Filters and SPLDA classifiers without an explicit negative set, depicted in Figure 4.3.

4.4 Solving Toeplitz Equations

Unfortunately, Toeplitz matrices are not diagonalised by the Fourier transform as circulant matrices are. There is, however, an extensive and varied body of literature surrounding the solution of Toeplitz systems. Some key

results are reviewed here.

4.4.1 Direct Methods

For general $\ell \times \ell$ matrices, factorisation algorithms take $O(\ell^3)$ time and solutions can subsequently be obtained in $O(\ell^2)$ time. One-level $\ell \times \ell$ Toeplitz matrices can instead be factorised in $O(\ell^2)$ time using Levinson recursion [38, 62], with solutions then obtained using the Gohberg-Semencul formula [25] in $O(\ell \log \ell)$ time. This is entirely without inflicting the $O(\ell^2)$ memory requirement of instantiating the explicit matrix or its inverse. Levinson recursion can be applied to *block* one-level Toeplitz matrices with $k \times k$ blocks to compute a factorisation in $O(k^3\ell^2)$ time [2]. Unfortunately, Levinson recursion cannot take advantage of two-level Toeplitz structure in general [69], although a handful of obscure exceptions have been identified [63, 69].

4.4.2 Iterative Methods

Iterative methods for minimising quadratic objectives generally involve a matrix-vector product to compute the gradient

$$\frac{\partial}{\partial x}\left(\frac{1}{2}x^T Ax - b^T x\right) = Ax - b \ . \tag{4.14}$$

While the Fourier transform cannot be used to directly invert a Toeplitz matrix, it can be used for rapid evaluation of matrix-vector products. Any Toeplitz matrix is a sub-matrix of some circulant matrix that is at least twice its size. Therefore a Toeplitz matrix-vector product can be computed as a larger circulant matrix-vector product using the FFT as described in Section 2.12. The input vector is padded with zeros and the output vector contains some unused values.

This fast multiply routine has been the motivation for a number of works which consider the Preconditioned Conjugate Gradient method (PCG) for solving block or multi-level Toeplitz equations. Since the objective function is smooth and strongly convex, simple gradient descent has linear convergence, requiring $O(\ln(1/\epsilon))$ iterations to achieve ϵ accuracy. However, the Conjugate Gradient method is generally preferred because it is capable of attaining super-linear convergence with similar computational effort per iteration, in particular if the eigenvalues of the matrix are clustered [46].

Instead of $Ax = b$, PCG considers the equivalent problem $MAx = Mb$, where the preconditioner M must be full rank and MA has more desirable spectral properties than A alone. PCG does not need the matrix M to be instantiated explicitly, it only requires that matrix-vector products can be computed. The ideal choice is $M = A^{-1}$, however to multiply a vector by this matrix is to solve the original problem.

A logical choice of preconditioner for Toeplitz systems of equations is the inverse of a similar circulant matrix, since matrix-vector products can be computed efficiently using the Fourier domain $(M^{-1}x)[u] = \hat{m}[u]^{-1}\hat{x}[u]$. Strang [59] originally proposed a circulant matrix which used only the inner diagonals of the Toeplitz matrix and was shown to guarantee super-linear convergence for a large class of problems [11]. Chan [12] instead considered the nearest matrix in the Frobenius sense

$$\arg\min_{M} \|M - A\|_F \quad \text{s.t.} \quad M \text{ is circulant}, \tag{4.15}$$

and observed that it was more effective at reducing the condition number and producing a matrix with clustered eigenvalues. A preconditioner with super-linear convergence renders the number of iterations a small constant, effectively yielding the solution to an $\ell \times \ell$ Toeplitz system in $O(\ell \log \ell)$ time.

Two-level circulant preconditioners have been proposed for two-level Toeplitz

systems [10] as well as block Toeplitz systems [13]. Serra Capizzano and Tyrtyshnikov [57] presented the theoretical result that multi-level circulant preconditioners are not guaranteed super-linear convergence for multi-level Toeplitz matrices by the same mechanism as one-level Toeplitz matrices, noting that fast convergence is still possible in practice.

Somewhat surprisingly, the block two-level circulant matrix in (4.13), which defines the linear system of equations for a Correlation Filter, is in fact the nearest in the Frobenius sense to the block two-level Toeplitz matrix of SPLDA, and is the solution to (4.15). Therefore, the procedure of training a Correlation Filter by solving a circulant system can be employed as a preconditioner within PCG to solve a Toeplitz system.

To summarise, the linear system defined by a symmetric, positive-definite, block multi-level Toeplitz matrix can be solved using Cholesky factorisation or the Conjugate Gradient method, optionally employing a circulant inverse preconditioner.

4.4.3 Time and Memory Complexity

Figures 4.4 and 4.5 show the dependence on the window size of the time and memory demands of different algorithms for solving the necessary linear systems of equations. These graphs were produced for the 31-channel HOG features described in the following section. The linear system was constructed from the covariance of real images and the detector was trained for a single positive window that was randomly sampled from a real image. The algorithms compared were: direct solution of the circulant system as per Multi-Channel Correlation Filters, direct solution of the Toeplitz system using Cholesky factorisation, iterative solution of the Toeplitz system using CG and PCG with a circulant preconditioner.

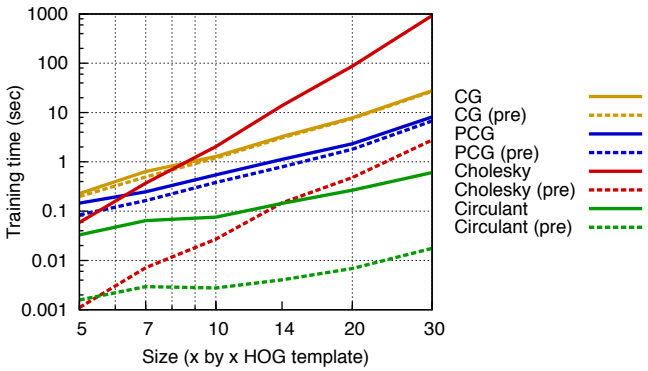


Figure 4.4: Empirical time taken to solve the linear system versus feature image size. The circulant algorithm and iterative Toeplitz algorithms scale much better than Cholesky factorisation. Times are reported including and excluding ("pre") all pre-computable factorisations and transforms.

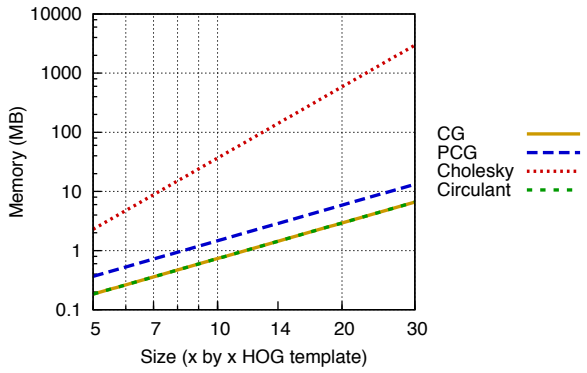


Figure 4.5: Theoretical memory requirement to solve the linear system versus feature image size. The circulant and iterative Toeplitz algorithms do not instantiate the full matrix and therefore scale linearly $O(m)$ instead of quadratically $O(m^2)$ in the number of pixels m.

The two direct algorithms were the fastest for small windows. The direct solution of the circulant system dominated all approaches in time and memory, however the circulant covariance matrix may introduce spurious correlations due to the implicit periodic extension of training images. This effect is quantified in the following section. As the window size grows, Cholesky factorisation becomes relatively slow unless the factorisation can be pre-computed. However, the amount of memory required to store this factorisation also grows rapidly with the window size (quickly reaching gigabytes), making it impractical to pre-compute the Cholesky factorisation for a number of different window sizes. This large memory requirement may also render the use of a pre-computed Cholesky factorisation infeasible in scenarios with limited memory such as on mobile devices. In situations where moderately-sized detectors need to be trained, and either memory is limited or detectors of several different sizes need to be trained, the circulant or iterative Toeplitz solutions would be preferred. The variant of the iterative method that adopts a preconditioner is faster and only requires about twice as much memory.

4.5 Pedestrian Detection with HOG Features

The performance of the detectors produced by Correlation Filters and SPLDA will be assessed in two canonical pedestrian detection benchmarks, the INRIA Person dataset [15] and the Caltech Pedestrian dataset [18]. Pedestrians are an appealing class to consider for sliding-window classification because they tend to occur with similar appearance and aspect ratio, enabling the use of a single template. The standard evaluation tool of Dollár et al. [18] will be used to enable meaningful comparisons to other work. This tool produces

Detection Error Tradeoff (DET) graphs that plot the miss rate (false negative rate) of a detector against its number of false positives per image (false positive rate) as the score threshold varies. The overall quality of a detector is summarised in the log-average miss rate for the operating range of 0.01 to 1 false positive per image. Previous evaluations of either method [29, 31, 34] have not adopted this standard evaluation metric for pedestrian detection.

The experiments aim to compare the Toeplitz and circulant algorithms to Hard Negative Mining with a Linear SVM, as well as to each other. Although the evaluation tool includes the output of the original HNM implementation [15] for reference, this algorithm has been re-implemented using the same routines for the feature transform, multi-scale search and Non-Maxima Suppression (NMS) as the Toeplitz and circulant implementations. This enables direct comparison of the results without having to consider these factors of variation. Furthermore, if the performance of the re-implementation matches the reference, then it validates the methodology to some degree. The result of training an SVM on a densely sampled set of windows was also included for comparison.

4.5.1 Implementation Details

The non-linear feature map used in the following experiments was the reference implementation of Histograms of Oriented Gradient (HOG) features provided by Felzenszwalb et al. [21] in their `voc-release5` package. A slight modification was made that removed boundary effects to ensure that the translation commutative property in (3.2) holds.

The following parameters were adopted for extracting examples and performing multi-scale sliding-window search:

- windows are of size 41×100 plus an extra 18 pixels of context included

on all sides,

- positive examples are resized so that their heights match, and any example whose aspect ratio is more than a multiplicative factor of 1.5 from the desired aspect ratio is excluded,

- NMS deems two windows to overlap if the ratio of the area of their intersection to the area of the smaller window is more than 0.65,

- multi-scale search with a geometric step of 1.07,

- a maximum search scale of $1\times$ for INRIA and $2\times$ for Caltech.

The HOG implementation also has a single configuration parameter, the length of the sides of the histogram cells, that was fixed to either four or eight pixels.

Cross-validation with five folds was used to choose the parameters of each algorithm before re-training a detector on the full training set to evaluate on the testing set. The extensive list of parameters for HNM is enumerated in Section 3.4. The number of initial negative examples was fixed at 10^4. The Toeplitz and circulant algorithms are far simpler, their only parameter being the regularisation weight λ. The mean and covariance of natural images under the feature transform were estimated from the full training set.

Algorithms were implemented in Go, making use of FFTW and LAPACK where appropriate. The source code is available online at `http://github.com/jvlmdr/shift-invar/`.

It should be noted that the evaluation tool of Dollár et al. uses a modified version of the INRIA Person testing set, in which the bounding boxes have been coerced to an aspect ratio of width/height = 0.41.

4.5.2 Results

The performance of detectors trained using each algorithm, with parameters chosen by cross-validation on the training set, are presented for the INRIA Person and Caltech Pedestrian datasets in Figure 4.6. These results were obtained using HOG cells of 8×8 pixels. Both experiments used the training data of the INRIA Person training set. Plots were generated using the toolbox of Dollár et al. [18]. It is standard practice to train a detector for the Caltech Pedestrian dataset using the INRIA Person dataset to confirm the cross-dataset generalisation ability of an algorithm.

The problem with using a *single detector* to evaluate the performance of a training *algorithm* is that it does not account for the variance in sampling the training and the testing set from their true underlying distributions. Cross-validation obtains multiple estimates of the generalisation error (one per fold), and these can be used to reduce the variance of the estimate by taking their mean, as well as to estimate the variance itself. To achieve this for the testing set, it was partitioned into mutually exclusive subsets, and each detector from cross-validation was evaluated on one partition. The results for both INRIA Person and Caltech Pedestrian datasets with variance estimates are shown in Figure 4.7.

4.5.3 Discussion

The re-implementation of HNM outperforms the baseline implementation in both experiments. This may be largely attributed to the fact that parameters in the original implementation were chosen to minimise a different error metric (miss rate at 10^{-4} false positives per window), and perhaps also to the use of different implementation of the HOG transform.

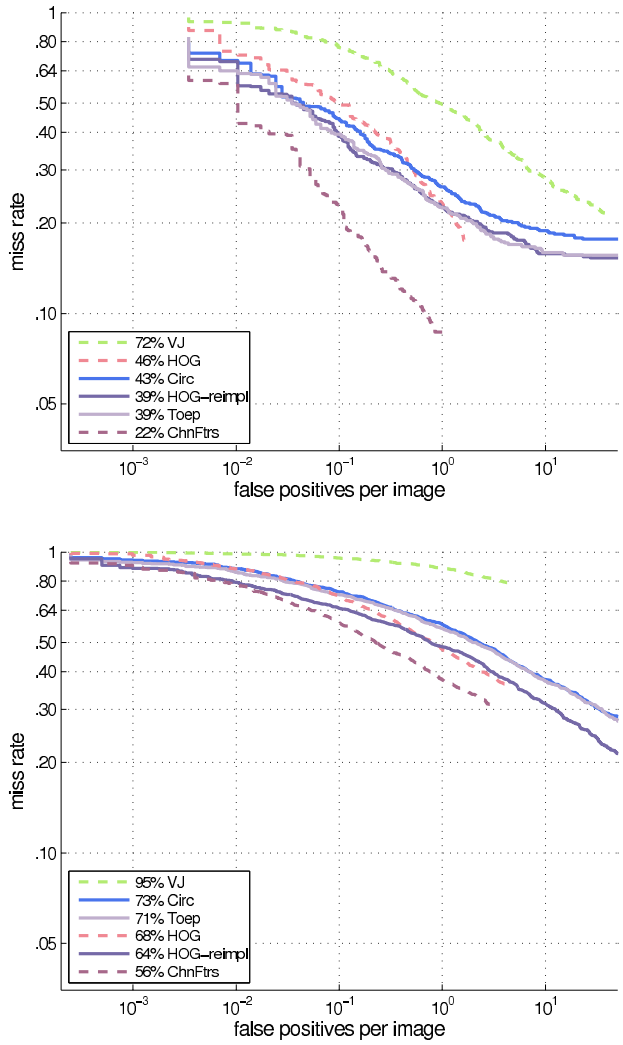


Figure 4.6: The Detection Error Tradeoff graph for pedestrian detectors trained on the INRIA Person training set and tested on the INRIA testing set (top) and the Caltech USA "Reasonable" testing set (bottom). The single number in the legend is the geometric average of the miss rate (lower is better) between 0.01 and 1 FPPI. *HOG* refers to the original HNM implementation of Dalal and Triggs [15] and *HOG-reimpl* refers to the re-implementation of this method. *Toep* and *Circ* are the Toeplitz and circulant algorithms. *VJ* and *ChnFtrs* denote the boosted decision tree algorithms of Viola and Jones [67] and Dollár et al. [17].

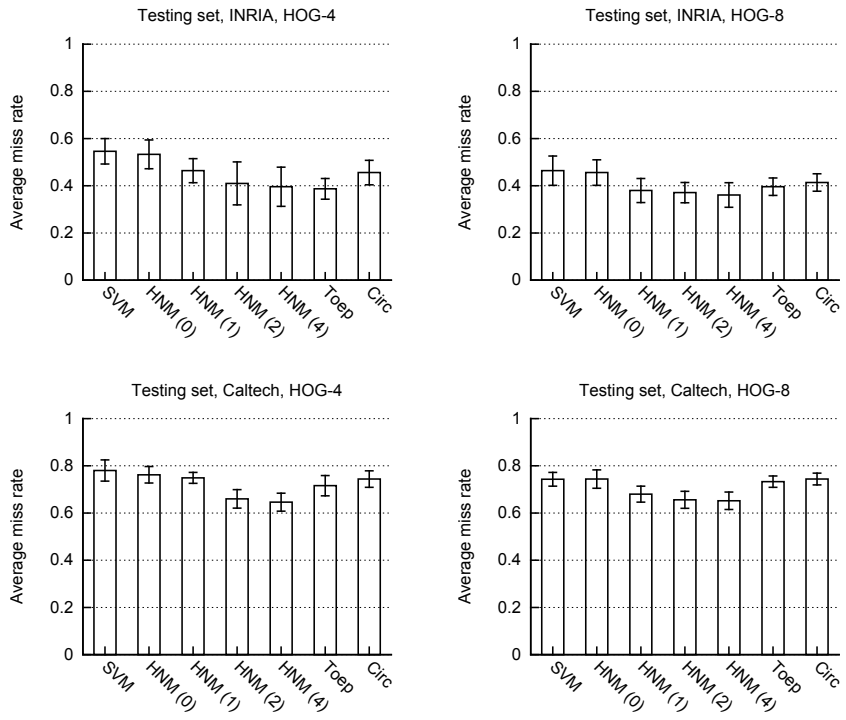


Figure 4.7: Distributions of log-average miss rates (lower is better) for the INRIA Person (top) and Caltech Pedestrian (bottom) testing sets using HOG features with four- (left) and eight-pixel (right) cells. Both experiments use the INRIA Person training set for cross-validation and training data. The error bars indicate one standard deviation. The number that follows each HNM label denotes the number of rounds of mining. Zero rounds of mining simply corresponds to an SVM trained on a random subset of negative windows. An SVM that was trained using every window on a densely sampled grid was also included for comparison.

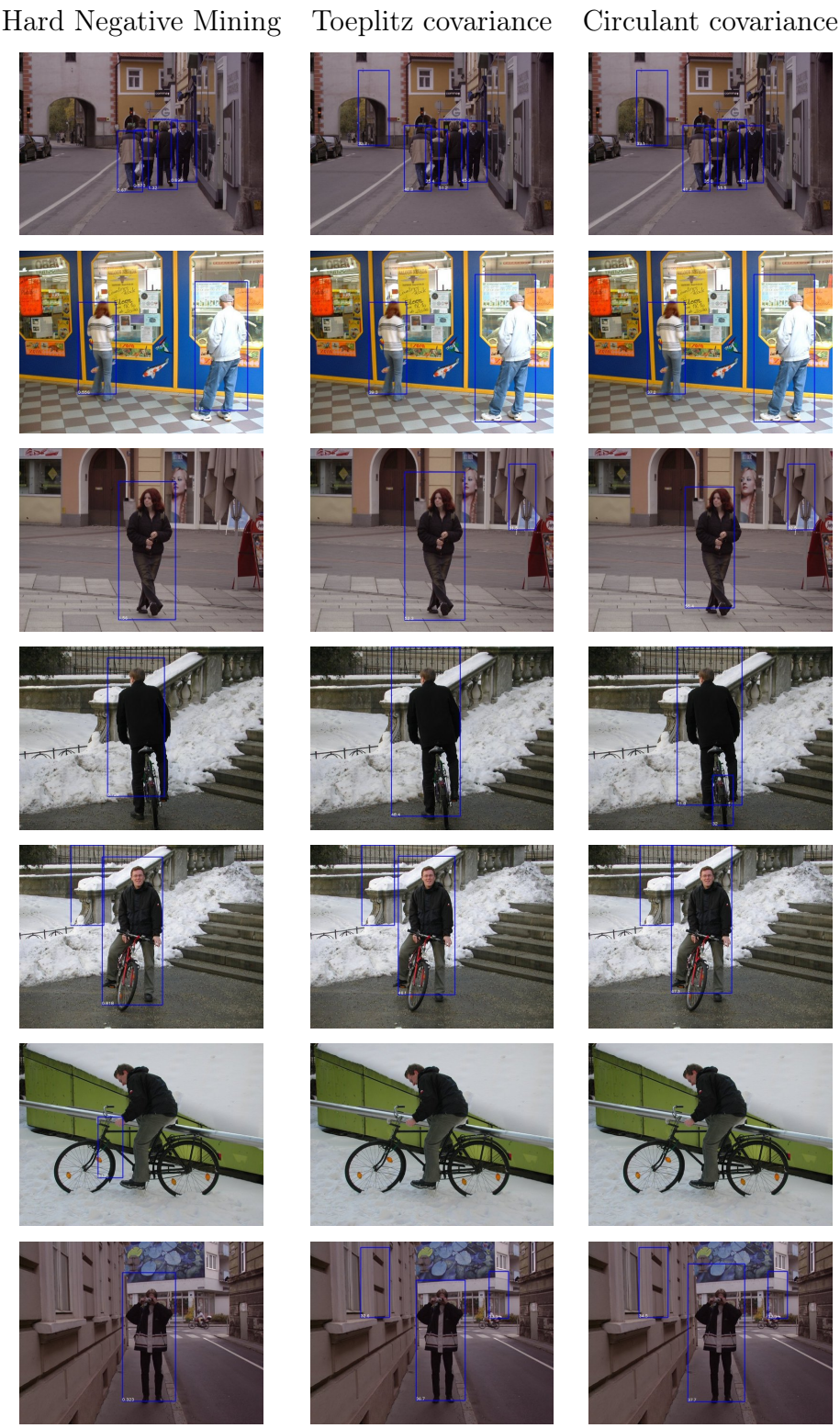
The Toeplitz algorithm is sometimes better (but never worse) than the circulant algorithm, and multiple rounds of Hard Negative Mining is sometimes better (but never worse) than the Toeplitz algorithm. The difference between the methods is sometimes insignificant: it is small compared to the variance.

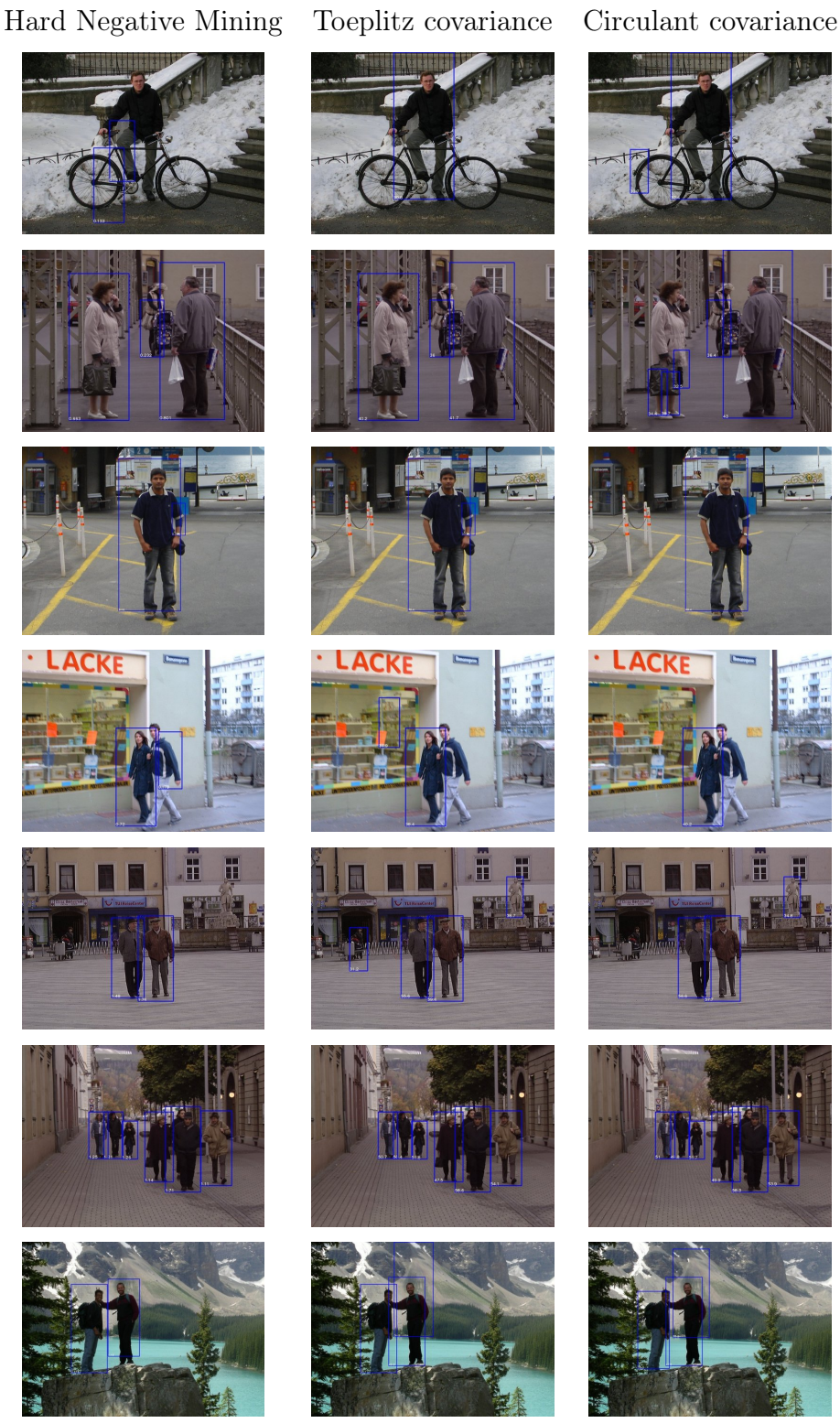
Unfortunately, none rival the performance of the boosted decision tree baseline of Dollár et al. [17]. However, the purpose of these experiments is not to demonstrate state-of-the-art detection performance, but to investigate the effectiveness of fast algorithms for training a linear detector. The detector of Viola and Jones [67] was also included by convention.

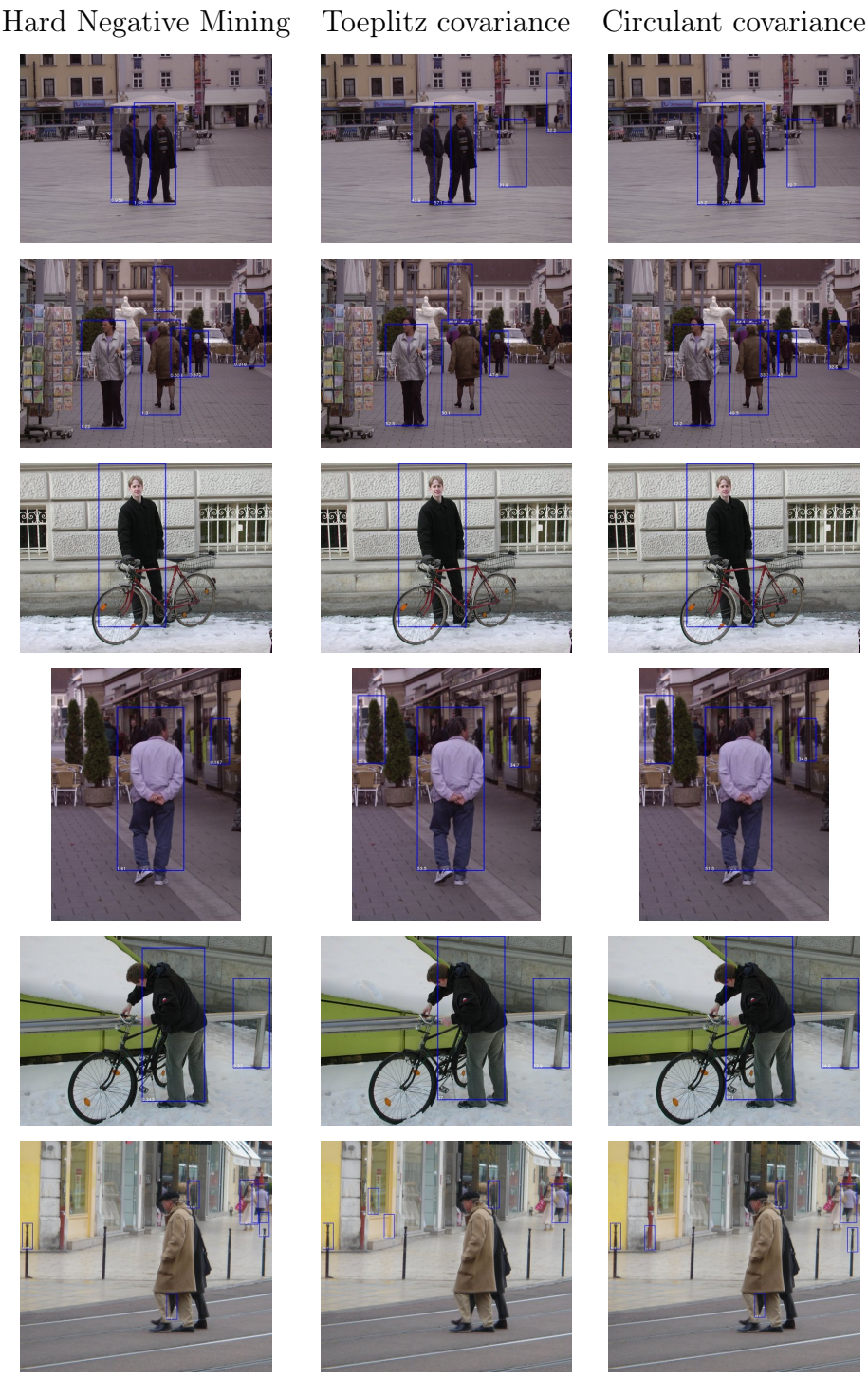
4.5.4 Example Detections

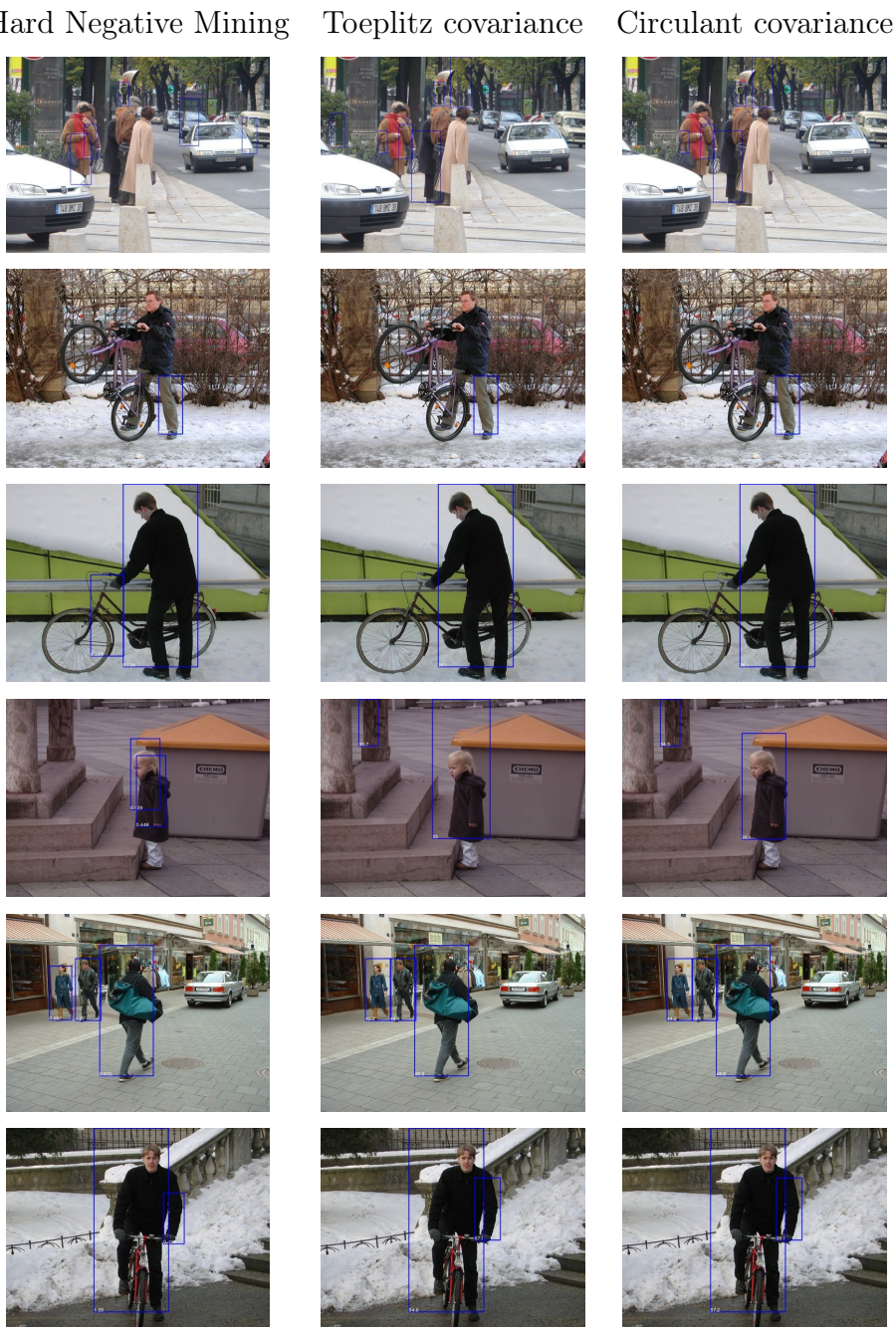
The following pages show the results of each detector for a set of random images in the INRIA testing set. The operating point was chosen to be one false positive per image.


Hard Negative Mining
Toeplitz covariance
Circulant covariance


Hard Negative Mining
Toeplitz covariance
Circulant covariance


Hard Negative Mining
Toeplitz covariance
Circulant covariance
LACKE
LACKE
LACKE


Hard Negative Mining
Toeplitz covariance
Circulant covariance


Hard Negative Mining
Toeplitz covariance
Circulant covariance

4.5.5 Performance versus Training Time

Figure 4.8 plots the miss rate of a detector against the time required to train it. The closed-form linear methods are significantly faster than using Hard Negative Mining while still achieving comparable performance.

Results are shown using HOG features that have cells of either 4×4 or 8×8 pixels. This parameter strongly impacts the running time because the dimensionality of the feature image using cells of 4×4 pixels is roughly four times that of the feature image using cells of 8×8 pixels (smaller cells imply a higher sample rate, a factor of two per dimension).

For the smaller template that uses eight-pixel HOG cells, regardless of the method that is used to solve the linear system, the time required to solve it is negligible compared to the setup time. To set up requires to load the covariance matrix from disk, to sample and resize the positive examples, and to compute their feature transform. For the larger template that uses four-pixel HOG cells, the Cholesky factorisation is significantly more expensive than the iterative algorithms or the circulant approach.

In some scenarios, the time to load the statistics from disk and compute feature transforms of the training data can be ignored, for example when training many detectors from a handful of examples each. Figure 4.9 plots the miss rate of a detector against the amount of time required just to solve the relevant linear system. Times are shown with and without pre-computable factorisations and transforms included. For the smaller template that uses eight-pixel HOG cells, the time required by the Cholesky factorisation is no more than that of the iterative methods or indeed the circulant method. However, for the larger template that uses four-pixel HOG cells, the time required to compute the Cholesky factorisation is significantly greater than to solve the Toeplitz equation using the Conjugate Gradient methods. This

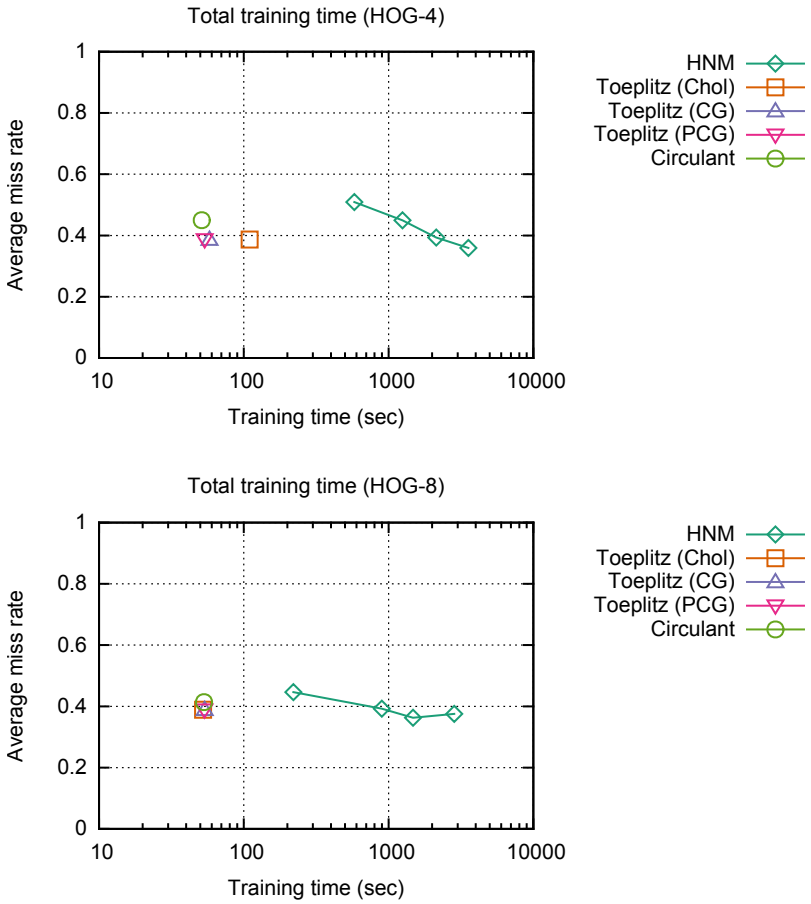


Figure 4.8: Miss rate (lower is better) versus training time using HOG features with four-pixel (top) and eight-pixel (bottom) cells. This experiment measures cross-validation error in the INRIA Person dataset. The markers for Hard Negative Mining indicate 0, 1, 2 and 4 rounds from left to right.

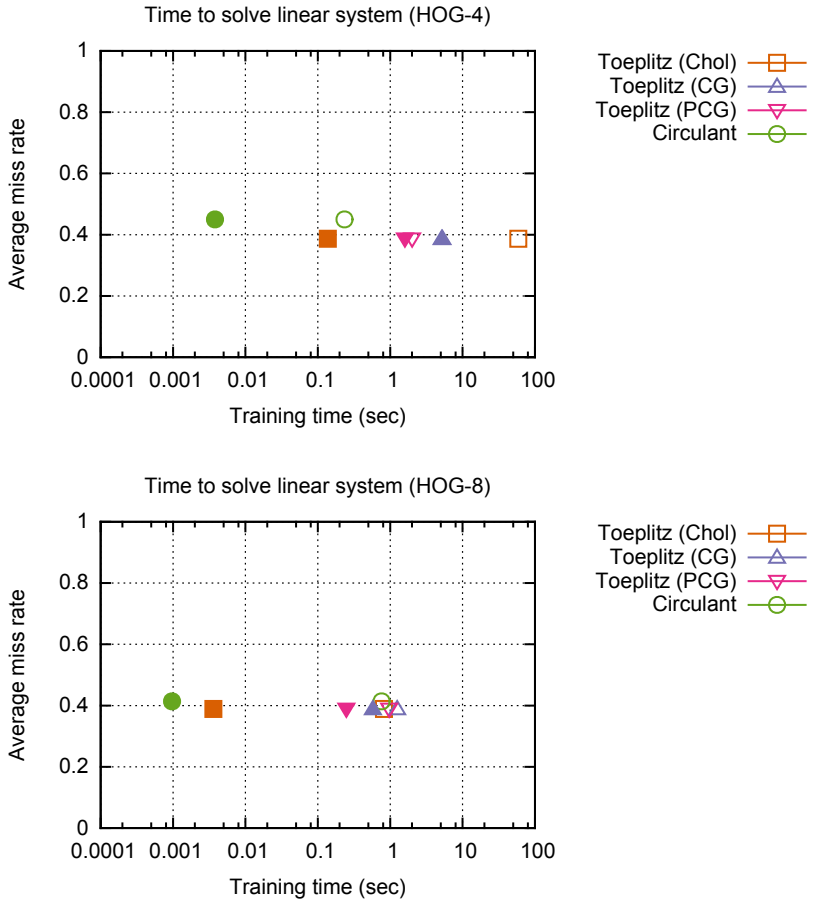


Figure 4.9: Miss rate (lower is better) versus time to solve the linear system for different algorithms using HOG features with four-pixel (top) and eight-pixel (bottom) cells. This experiment measures cross-validation error in the INRIA Person dataset. Times are reported including (hollow markers) and excluding (solid markers) all pre-computable factorisations and transforms.

is not the case when the factorisation can be pre-computed and cached. However, as established in the previous section, this incurs a large memory overhead. The circulant system is always the fastest to solve, sometimes by orders of magnitude, although it can incur an increase in the miss rate.

4.5.6 Banded Toeplitz Covariance

If two (feature) pixels are far enough from one another in an image, then they may be independent and therefore have zero correlation. This suggests that the Toeplitz covariance matrix might be banded such that $S_{pq}[u,t] = 0$ unless $\|u-t\|_\infty < b$, where b will be referred to as the bandwidth. The smaller the bandwidth, the less memory required to store the covariance matrix and the less time required to estimate it. This section will undertake an empirical investigation to determine the effect that a banded Toeplitz covariance matrix has on the detection performance.

Imposing a bandwidth on a Toeplitz covariance matrix $S \succeq 0$ by setting some entries to zero can be considered element-wise multiplication $S \odot M$ by a zero-one Toeplitz mask matrix $M[u,t] = m[u-t]$ that has elements

$$m_{pq}[\delta] = \begin{cases} 1 & \text{if } \|\delta\|_\infty < b, \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

For least-squares regression and LDA, the covariance matrix must be positive semidefinite. The Schur product theorem states that if $A \succeq 0$ and $B \succeq 0$, then $A \odot B \succeq 0$. Therefore, while it is not necessary that the mask be positive semidefinite $M \succeq 0$, it would at least guarantee that the product is positive semidefinite.

The zero-one bandwidth mask is shown not to be positive semidefinite in

general via a simple counter-example

$$\lambda\left(\begin{bmatrix}1 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 1\end{bmatrix}\right) \approx (1.41, 1, -0.41) \ . \tag{4.17}$$

An alternative Toeplitz mask which is positive semidefinite is the Gaussian mask $G[u,t] = g[u-t]$ with elements

$$g_{pq}[\delta] = \exp\left(-\frac{1}{2}\|\delta\|^2/\sigma^2\right) \ . \tag{4.18}$$

This matrix is known to be positive semidefinite because it is the Fourier series of a positive periodic function. Therefore, an appropriate practical mask of S is $(M \odot G) \odot S$, since the elements of $G \odot S$ will be approximately zero at sufficient distance from the diagonal.

Figure 4.10 presents the empirical effect of different choices of b and σ in forming G. Swathes of the graph are missing because σ was too large compared to b and the resulting matrix was indefinite. A sensible choice is $b \approx 2.5\sigma$, corresponding to about 98.7% of the integral of a normal distribution. The bandwidth can only be slightly reduced before it is necessary to choose σ small such that the performance of the resulting detector is significantly worse. The minimum bandwidth before the performance is affected is about 80 pixels (twenty four-pixel HOG cells or ten eight-pixel HOG cells).

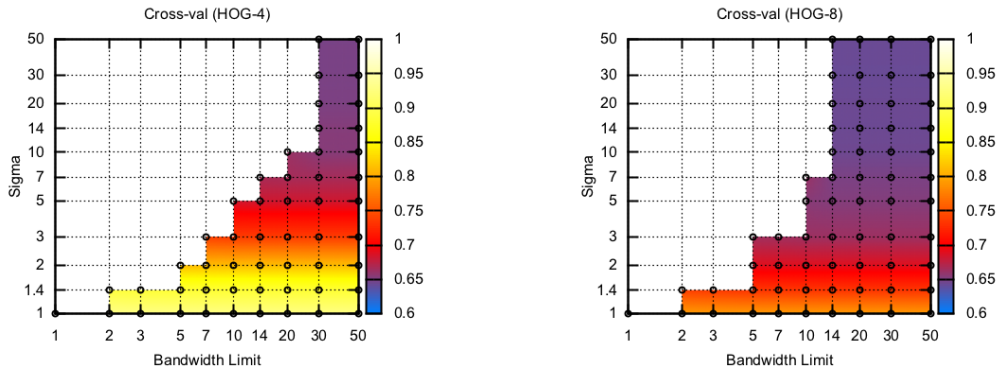


Figure 4.10: Detection error (lower is better) for different choices of bandwidth and Gaussian envelope parameter σ. When imposing a hard limit on the bandwidth of the Toeplitz covariance matrix, it is necessary to multiply the matrix by a Gaussian envelope with isotropic variance σ to ensure that the matrix remains positive semidefinite. The configuration at every grid point was evaluated: black markers indicate that the matrix was positive definite and a detector could be obtained, and the absence of a marker indicates an indefinite matrix. Larger σ results in a higher error rate. At the maximum bandwidth (the right of each graph), the bandwidth exceeds the size of the window. This experiment measures detection error as the log-average miss rate for cross-validation on the INRIA Person training set using HOG features with 4-pixel and 8-pixel cells.

Chapter 5

Trajectory-Based Non-Rigid Structure-from-Motion

5.1 Problem Description

Structure-from-Motion (SfM) is the problem of recovering the 3D structure of an object from multiple 2D images. There must be some apparent 3D motion between the images, or else the positions of the projected points will be identical, and geometric reconstruction from a single image is impossible. The canonical SfM problem assumes that the object is rigid, which means that the 3D distance between every pair of points is constant across all images. This is true either when the images are captured at the same instant by multiple cameras or when a single camera roams a static scene.

Non-Rigid Structure-from-Motion (NRSfM), on the other hand, permits the object to change shape between images. Without some additional constraints on the structure, this problem devolves to independent single-view reconstruction in each image. While some works consider the general problem of non-rigid reconstruction from an unordered set of images, this thesis

will concentrate on trajectory-based methods, for which the images must comprise consecutive frames of video.

5.2 Formulation

It will be assumed that a set of 2D points that are in correspondence have been identified in each image, for example by matching descriptors or tracking frame to frame. The shape of the object is the 3D positions of these points.

First consider the rigid problem where there are n points and ℓ images. Each image $t = 0, \ldots, \ell - 1$ contains the projections $w_{ti} \in \mathbb{R}^2$ for a subset of points that are visible $i \in \mathcal{V}_t \subseteq \{1, \ldots, n\}$. Let the 2D projection of a 3D point $x \in \mathbb{R}^3$ in image t be determined by the function $P_t(x, \xi_t)$ with unknown camera parameters $\xi_t \in \mathcal{Q}$. The problem is then to find the points $x = (x_1, \ldots, x_n) \in \mathbb{R}^{3n}$ and cameras $\xi = (\xi_0, \ldots, \xi_{\ell-1}) \in \mathcal{Q}^\ell$ that minimise the total projection error

$$\begin{aligned} \arg\min_{\xi, x} \quad & \sum_{t=0}^{\ell-1} \sum_{i \in \mathcal{V}_t} f_t(x_i, \xi_t; w_{ti}) \\ \text{subject to} \quad & \xi_t \in \mathcal{Q}, \qquad t = 0, \ldots, \ell - 1 \ . \end{aligned} \tag{5.1}$$

This may be a constrained optimisation problem due to the parameterization of camera pose, possibly using rotation matrices or unit quaternions. The projection error of point i in image t is measured using $f_t(x_i, \xi_t; w_{ti})$. Ideally this is some non-decreasing function $\rho : [0, \infty) \to \mathbb{R}$ of the Euclidean norm of the difference

$$f_t(x_i, \xi_t; w_{ti}) = \rho(\|w_{ti} - P_t(x_i, \xi_t)\|) \ , \tag{5.2}$$

although other loss functions may be considered for the sake of optimisation.

The problem of non-rigid reconstruction from ℓ frames of video demands to solve for the time-evolving shape $x = (x_1, \ldots, x_n)$ described by the 3D

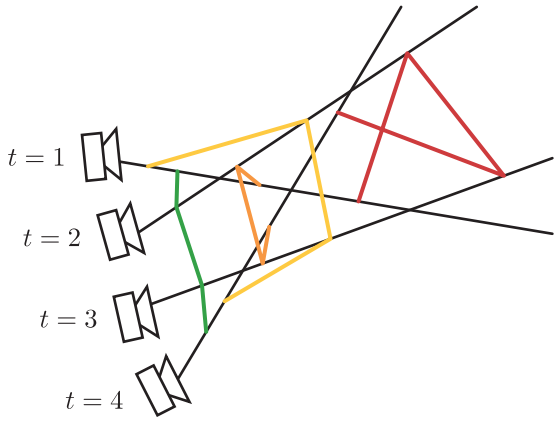


Figure 5.1: The back-projected rays through an observed point from a known moving camera define a ray of infinite solutions for the position of each point per frame. However, experience suggests that trajectories are more likely to be slow and smooth (green) than fast and erratic (yellow to red). Trajectory-based methods either define a likelihood over the space of all trajectories, or restrict the solution to a region of the space.

trajectory of each point $x_i : \mathcal{U} \to \mathbb{R}^3$ in all frames $\mathcal{U} = \{0, \ldots, \ell - 1\}$, that minimise projection error

$$\begin{aligned} &\arg\min_{\xi, x} \quad \sum_{t=0}^{\ell-1} \sum_{i \in \mathcal{V}_t} f_t(x_i[t], \xi_t; w_{ti}) \\ &\text{subject to} \quad \xi_t \in \mathcal{Q}, \qquad t = 0, \ldots, \ell - 1 \ . \end{aligned} \tag{5.3}$$

If the 3D structure were known, then the problem of solving for the cameras alone would be identical in the rigid and non-rigid cases. The key difference between the problems is that if the cameras were known, then in the rigid case each point could be determined by multi-view triangulation, whereas in the non-rigid case there would still be an infinite set of solutions, illustrated in Figure 5.1.

Therefore, the problem that will be considered is the reconstruction of a

deforming 3D shape where the cameras are known. In practice, a set of cameras may be obtained by applying rigid SfM to a rigid subset of the scene [50], using the nuclear-norm method of Dai et al. [14] or simply attempting to solve rigid SfM where the non-rigid component is treated as noise.

5.3 Reconstruction with a Trajectory Basis

5.3.1 Background

A deformable structure is a set of 3D shapes. Bregler et al. [8] introduced the assumption that there exists a low-dimensional subspace that is close to every element of the set, and hence every possible configuration of the structure is well-approximated by a linear combination of a few basis shapes. Inspired by the bi-linear factorisation algorithm developed by Tomasi and Kanade [61] for rigid SfM under weak-perspective projection, they proposed a tri-linear factorisation algorithm to recover the camera pose, the set of basis shapes, and the basis coefficients for each image.

Akhter et al. [3] later recognised that when the images are ordered frames of video, the shape subspace constraint has a dual interpretation as a *trajectory* subspace constraint due to the equal dimension of the row- and column-spaces of a matrix. The advantage of considering a trajectory subspace is that a generic basis for continuous functions can be used, eliminating the need to solve for sequence-specific basis vectors. This reduced the problem of jointly solving for cameras and non-rigid structure from tri-linear factorisation to bi-linear. The generic basis typically consists of the low frequency basis vectors of the Discrete Cosine Transform (DCT), with an independent identical basis for each of the three dimensions. The DCT is chosen for its ability to compactly represent natural signals [3].

Park et al. [50] recognised that when the cameras are known, reconstruction using a trajectory basis is an independent problem per trajectory. Henceforth only a single point's trajectory will be considered.

5.3.2 Formulation

Let $x : \mathcal{U} \to \mathbb{R}^3$ be a signal that defines the trajectory of a 3D point, with elements $x_p[t] \in \mathbb{R}$ for dimensions $p = 1, 2, 3$ and samples $t \in \mathcal{U} = \{0, \ldots, \ell - 1\}$. If each coordinate $x_p : \mathcal{U} \to \mathbb{R}$ is constrained to the subspace defined by $k \leq \ell$ basis functions $\phi_j[t]$ for $j = 1, \ldots, k$, then the trajectory is sufficiently specified in basis coefficients β_{pj} according to

$$x_p[t] = \sum_{j=1}^{k} \beta_{pj}\phi_j[t], \qquad p = 1, 2, 3, \;\; t = 0, \ldots, \ell - 1 \;\; . \tag{5.4}$$

This can be expressed $x_p = \Phi\beta_p$ where Φ is an $\ell \times k$ matrix whose columns are ϕ_j, or simply $x = \Theta\beta$ where Θ is $3\ell \times 3k$. It can be assumed without loss of generality that the basis vectors are orthonormal $\Phi^T\Phi = I$ and $\Theta^T\Theta = I$.

Under a perspective camera model, the non-linear projection function is

$$P_t(x, \xi_t) = \frac{\left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right] K_t(R_t x + d_t)}{\left[\begin{smallmatrix} 0 & 0 & 1 \end{smallmatrix}\right] K_t(R_t x + d_t)} \tag{5.5}$$

where R_t is a 3×3 matrix giving the camera orientation, $d_t \in \mathbb{R}^3$ defines the camera position and K_t is a 3×3 matrix that describes the intrinsic camera calibration [30]. If the loss function computes the Euclidean norm of the projection error

$$f_t(x, \xi_t; w) = \|w - P_t(x, \xi_t)\| \tag{5.6}$$

(or the squared Euclidean norm), then it is not convex in x since projection involves division by the depth. A common technique in multi-view triangulation is to scale the projection residual by the depth to obtain a linear

expression. Let A_t, b_t, ν_t and ζ_t be a partition of the 3×4 matrix

$$K_t \begin{bmatrix} R_t & d_t \end{bmatrix} = \begin{bmatrix} A_t & b_t \\ \nu_t^T & \zeta_t \end{bmatrix} \quad . \tag{5.7}$$

The projection error can then be expressed

$$r = \frac{1}{z_t(x)}(A_t x + b_t) - w \tag{5.8}$$

with $z_t(x) = \nu_t^T x + \zeta_t$ and the depth-scaled error is a linear expression

$$z_t(x)\, r = Q_t x - u_t \tag{5.9}$$

where $Q_t = A_t - w\nu_t^T$ is a 2×3 matrix and $u_t = \zeta_t w - b_t$. Adopting the squared Euclidean norm of the depth-scaled error, the total projection error over all frames is simply

$$\sum_{t=0}^{\ell-1} f_t(x[t], \xi_t; w_t) = \sum_{t=0}^{\ell-1} \|Q_t x[t] - u_t\|^2 = \|Qx - u\|^2 \tag{5.10}$$

where Q is $2\ell \times 3\ell$.

The projection equations are exactly satisfied if $Qx = u$. Under the trajectory subspace constraint, this becomes $Q\Theta\beta = u$, which is over-determined provided that $\operatorname{rank}(Q\Theta) > 3k$, for which it is necessary that $2\ell > 3k$. The solution that minimises the depth-scaled projection error is

$$x^\star = \Theta\beta^\star, \quad \beta^\star = \arg\min_\beta \|Q\Theta\beta - u\|^2 . \tag{5.11}$$

5.3.3 Reconstructability

Park et al. [50] observed that much more accurate reconstructions could be obtained when the camera moved significantly between frames. This accords with the intuitive notion that stereo reconstruction is more accurate with a greater baseline between cameras (although appearance matching may be

more challenging). They developed a measure of "reconstructability" that indicates how well the trajectory of a point $x : \mathcal{U} \to \mathbb{R}^3$ can be reconstructed from the observations of a pinhole camera whose optical centre moves with trajectory $c : \mathcal{U} \to \mathbb{R}^3$. Given an orthonormal basis Θ for the trajectory subspace $\text{col}(\Theta)$, reconstructability is defined as the ratio of the orthogonal component (to the subspace) of the camera trajectory to that of the point trajectory

$$\eta(x, c, \Theta) = \frac{\|(I - \Theta\Theta^T)c\|}{\|(I - \Theta\Theta^T)x\|} \quad . \tag{5.12}$$

The operator $I - \Theta\Theta^T$ is a projector for the space complementary to the trajectory subspace. It is important that the camera centre does not lie on the subspace because it, too, satisfies projection constraints. If the reconstruction is exact, then $\eta \to \infty$ since the point trajectory lies on the subspace $x \in \text{col}(\Theta)$ and the camera trajectory does not $c \notin \text{col}(\Theta)$. Park et al. [50] established that, conversely, if $\eta \to \infty$ and there exists a unique solution, then the reconstruction must be exact.

Chapter 6

Convolutional Prior in Non-Rigid Structure-from-Motion

6.1 Overview

Reconstructability [50] gives limited insight into the factors that affect the *accuracy* of a reconstruction because it merely describes a condition under which reconstruction is exact. The major contribution of this work is to better characterise the limitations of non-rigid reconstruction where only temporal relationships between variables are assumed. This is achieved through the development of a novel upper bound on 3D reconstruction error. Whereas reconstructability is only defined for reconstruction using a subspace constraint, the new bound is defined for reconstruction under the more general assumption that the trajectory is a Gaussian process with known covariance. This includes the subspace constraint as a special case where the precision matrix (inverse covariance matrix) is the projector on to the complementary subspace.

The new bound on reconstruction error highlights the importance of the

condition number of the system of linear equations that must be solved to obtain a reconstruction. A good solution is guaranteed if the value of the bound is small, and the bound can only become arbitrarily large if the system of equations is poorly conditioned. This is more likely to occur if the precision matrix of the Gaussian distribution has multiple eigenvalues that are very large compared to multiple other eigenvalues. The subspace approach is therefore fundamentally susceptible to being poorly conditioned: the corresponding precision matrix is a projector and therefore has a number of eigenvalues that are zero with the rest being one.

Motivated by the bound, alternative precision matrices are sought that are less likely to result in a poorly conditioned system of equations. This work introduces the assumption that the trajectory is a stationary process and therefore the precision matrix is Toeplitz, aside from boundary effects (it would be exactly Toeplitz were the trajectory defined on the set of all integers as described in Section 2.14). Adopting a Toeplitz precision matrix is appealing for trajectory reconstruction because the precision matrices for trajectories of different lengths are all specified at once.

It is proposed that near-Toeplitz precision matrices for trajectory reconstruction be obtained as the Gram matrix $\Lambda = G^T G$ of a Toeplitz operator that corresponds to convolution $Gx = x * g$. The semi-norm defined by the precision matrix is then equal to the norm of the response of the trajectory to some filter $x^T \Lambda x = \|x * g\|^2$. In fact, it is made apparent that stationarity was already implicit in the use of the DCT due to its connection with convolution. First- and second-difference filters have the physical significance of estimating the velocity and acceleration, and their corresponding matrices are shown to have desirable spectral properties. Trajectory reconstruction using these simple filters virtually eliminates the issue of the condition num-

ber and obtains a better solution than using a DCT subspace without the need to manually specify the optimal subspace dimension.

Finally, it is recognised that the compact support of first- and second-difference filters admits an efficient solution for combinatorial problems where the trajectory must belong to a finite set. This is applied to the problem of articulated trajectory reconstruction, which was previously solved using Branch and Bound [49].

6.2 Gaussian Trajectory Prior

Park et al. [50] restricted trajectories to a subspace $x = \Theta\beta$ to ensure a unique solution. The optimal trajectory was then chosen to minimise the affine projection error $\|Q\Theta\beta - u\|^2$. While this would practically be solved as an unconstrained least squares minimisation

$$\arg\min_{\beta} \quad \|Q\Theta\beta - u\|^2 \ , \tag{6.1}$$

it is equivalent to solving

$$\arg\min_{x} \quad \|Qx - u\|^2 \quad \text{subject to} \quad \Theta_\perp^T x = 0 \tag{6.2}$$

where $\Theta_\perp$ is an orthogonal basis for $\text{null}(\Theta^T)$ such that $\Theta^T\Theta_\perp = 0$ and $\Theta_\perp^T\Theta_\perp = I$. If Θ comprises the k lowest frequencies of a DCT basis of dimension ℓ, then $\Theta_\perp$ comprises the $\ell - k$ highest frequencies. Let $P_\Theta = \Theta\Theta^T$ be the projector on to the column-space of Θ, then the projector on to its complementary left nullspace is $I - P_\Theta = \Theta_\perp\Theta_\perp^T$.

Rather than use a subspace constraint to limit the degrees of freedom and ensure a unique solution, a regularisation term that penalises unlikely trajectories could be added to the objective. A convex quadratic penalty $\|x\|_\Lambda^2 = x^T\Lambda x$ with $\Lambda \succeq 0$ is an attractive option because the optimisation

problem is still the unconstrained minimisation of a least squares objective, and because it can capture a soft version of the subspace constraint by choosing $\Lambda = I - P_\Theta$. The modified objective is a combination of the projection error and this regulariser

$$\arg\min_x \quad \|Qx - u\|^2 + \|x\|^2_\Lambda \ . \tag{6.3}$$

This has the probabilistic interpretation of maximising the posterior likelihood $p(X|U) \propto p(U|X)p(X)$ of the trajectory X given noisy projections U and a prior distribution of trajectories $p(X)$ where the trajectory is a three-channel one-dimensional random process. The choice of a quadratic cost function corresponds to the negative log likelihood of a zero-mean Gaussian distribution with precision matrix $\Lambda = \Sigma^{-1}$.

For the sake of analysis, it is useful to consider the problem of finding the most likely trajectory that exactly satisfies the projection constraints

$$\arg\min_x \quad \|x\|^2_\Lambda \quad \text{subject to} \quad Qx = u \ . \tag{6.4}$$

This formulation may, in fact, be more desirable since the affine residual $\|Qx - u\|^2$ measures the depth-scaled projection error as described in the previous chapter, whereas the constraint $Qx = u$ guarantees that the true projection error is zero. This is the same reason that linear triangulation methods should only be used to provide initialisation for the minimisation of a non-convex cost function in triangulation [30].

6.3 Simulated Experiment

A synthetic experiment was established to quantify the accuracy of non-rigid reconstruction algorithms. In each trial of the experiment, a human motion sequence from the CMU MoCap dataset (`http://mocap.cs.cmu.edu/`) is

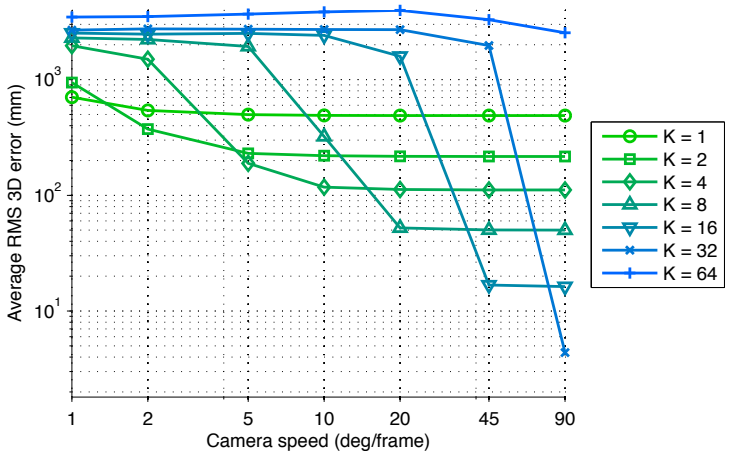


Figure 6.1: Reconstruction error versus orbiting camera speed for DCT bases of varying dimension k. The optimal basis dimension depends on the degree of motion of the camera. If this parameter were obtained from an oracle, then the reconstruction error would be a lower bound to these curves.

projected into the view of a perspective camera that orbits the scene on a horizontal plane at a constant rate. Sequences are 100 frames long and sampled at 30 frames per second. The reconstruction error of a trajectory is measured as the RMS 3D distance from ground truth over all frames. Results are averaged over all points in the skeleton and over 100 trials with different motion sequences.

The rate at which the camera orbits the scene was varied from 1 to 90 degrees per frame. Instinctively, a camera that orbits at 90 degrees per frame is expected to permit a better reconstruction than a slower rate, since each frame observes roughly the direction that was unobservable in the previous frame. This is confirmed in Figure 6.1, which plots the reconstruction error of the solution to (6.4) for scenes with different camera speeds, using DCT bases of different dimension k.

Figure 6.1 presents the results of the experiment using the formulation in (6.4). These results illustrate the critical nature of choosing the optimal basis dimension k. If it is chosen too small, then the basis will be unable to represent the trajectory well, but if it is chosen too large, then the system will be poorly conditioned and the solution extremely sensitive to noise. Noise is always present in practice, since the trajectory never lies *exactly* on the lower-dimensional subspace. When the camera moves faster, a larger basis dimension k can be used.

6.4 Reconstruction Error Bound

6.4.1 Criticism of Reconstructability

The intuition that a better reconstruction can be obtained with faster-moving cameras is reflected in the reconstructability function $\eta(x, c, \Theta)$ introduced by Park et al. [50], the definition of which is given in (5.12). Their function computes the ratio of camera motion to point motion that is orthogonal to the subspace. For perfect reconstruction to be achieved, this ratio must approach infinity: the denominator must be zero for the point trajectory to lie on the subspace, and the numerator must be non-zero since the optical centre satisfies projection constraints and would therefore also be a feasible solution if its motion lay on the subspace.

A severe criticism of this reconstructability measure, however, is that it does not capture the condition of the problem, which describes the sensitivity of its solution to noise. Even if the trajectory of the point lies exactly on the subspace and that of the camera does not, there may be many trajectories which are very close to both lying on the subspace and satisfying projection constraints. This can cause small perturbations of the observations to

manifest in large perturbations of the solution.

This is not the only problem with the measure. While it should approach infinity for a perfect reconstruction, its implications are unclear for an inexact reconstruction, where the point trajectory does not lie on the basis. Additionally, moving the camera centre along the line that connects it and the point in each frame does not affect the system of equations and therefore the solution, but does affect reconstructability. Finally, the dependence on the trajectory of the optical centre prohibits the analysis of affine cameras, which only have an optical centre at infinity. To remedy these issues, a theoretical upper bound on reconstruction error will be established.

6.4.2 Existence of a Unique Solution

Before establishing a bound on reconstruction error, the conditions for the problem in (6.4) to have a unique solution are examined.

While this problem can be solved using Lagrange multipliers, for the purpose of analysis instead let the feasible hyper-plane be parameterised

$$\{x : Qx = u\} = \{x_0 + Q_\perp z : z \in \mathbb{R}^\ell\} \ . \tag{6.5}$$

Here x_0 is any solution to $Qx_0 = u$ and $Q_\perp$ is a $3\ell \times \ell$ matrix whose columns are an orthogonal basis for null(Q) such that $QQ_\perp = 0$ and $Q_\perp^T Q_\perp = I$. Any solution to (6.4) can be expressed $x^\star = x_0 + Q_\perp z^\star$ where $z^\star$ is a solution to the unconstrained problem

$$z^\star = \arg\min_z \|x_0 + Q_\perp z\|_\Lambda^2 \ , \tag{6.6}$$

which is equivalent to the linear system of equations

$$(Q_\perp^T \Lambda Q_\perp) z^\star = -Q_\perp^T \Lambda x_0 \ . \tag{6.7}$$

Therefore there exists a unique solution to (6.4) if and only if $Q_\perp^T \Lambda Q_\perp$ is invertible.

This matrix is only invertible if the nullspace of the precision matrix Λ does not intersect that of the projection matrix Q except in the trivial space

$$\text{null}(\Lambda) \cap \text{null}(Q) = \{0\} \ . \tag{6.8}$$

This is proved in Section B.1. The physical meaning of this statement is that there does not exist a zero-cost trajectory that goes unobserved by the projection matrix. If such a trajectory existed, then any scalar multiple of it could be added to another solution without affecting its cost or violating projection constraints, and there would be infinite solutions. The nullspace of the precision matrix usually contains at least the constant non-zero trajectory (known as the "DC component" in electrical engineering) to avoid endowing the origin of the coordinate frame with significance.

When performing reconstruction using a trajectory subspace, the precision matrix is the projector on to the complementary subspace $\Lambda = I - P_\Theta$ and its nullspace is the trajectory subspace itself. In this regard, a subspace prior may be a poor choice since the precision matrix $\Lambda = I - P_\Theta$ will have a nullspace of dimension $3k$ for a coordinate-wise basis of dimension k. The subspace dimension k must be chosen to satisfy $3k \leq 2\ell$, otherwise the two nullspaces will have a non-trivial intersection since they are subspaces of $\mathbb{R}^{3\ell}$ and $3k + \ell > 3\ell$.

6.4.3 Upper Bound on Reconstruction Error

The condition number of a matrix determines how sensitive a system of equations is to noise. If $Ax = b$ and A is invertible, then the perturbed system $A(x + \delta x) = b + \delta b$ has its solution bounded in terms of the condition

number [60]

$$\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A)\frac{\|\delta b\|}{\|b\|} \ . \tag{6.9}$$

If the condition number is large, then a small perturbation of b could result in a large perturbation of x. For a symmetric positive semidefinite matrix A, the condition number is equal to the ratio of the maximum to the minimum eigenvalue

$$\text{cond}(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \|A\| \cdot \|A^{-1}\| \ . \tag{6.10}$$

This section will establish a bound on reconstruction error that incorporates the condition number of the matrix from the previous section.

Theorem 6.1. *If there exists a unique solution $x^\star$ to* (6.4)*, then its Euclidean distance from the true trajectory x is bounded $\|x - x^\star\| \leq \upsilon(x, Q, \Lambda)$ where*

$$\upsilon(x, Q, \Lambda) = \text{cond}(Q_\perp^T \Lambda Q_\perp)\frac{\|Q_\perp^T \Lambda x\|}{\|Q_\perp^T \Lambda Q_\perp\|} \ . \tag{6.11}$$

Proof. An expression for reconstruction error can be obtained by choosing x_0 to be the ground-truth trajectory $x_0 = x$

$$\|x - x^\star\| = \|Q_\perp z^\star\| = \left\|(Q_\perp^T \Lambda Q_\perp)^{-1} Q_\perp^T \Lambda x\right\| \ . \tag{6.12}$$

Of course the ground-truth trajectory is never known in practice, this is purely for the purpose of theoretical analysis. This expression facilitates the definition of an upper bound using the operator norm

$$\|x - x^\star\| \leq \|(Q_\perp^T \Lambda Q_\perp)^{-1}\| \cdot \|Q_\perp^T \Lambda x\| \tag{6.13}$$

$$= \text{cond}(Q_\perp^T \Lambda Q_\perp)\frac{\|Q_\perp^T \Lambda x\|}{\|Q_\perp^T \Lambda Q_\perp\|} \ . \tag{6.14}$$

□

6.4.4 Interpretation of the Bound

To understand the bound in (6.11), it helps to decompose it into the product of two constituent terms

$$\upsilon(x, Q, \Lambda) = \underbrace{\mathrm{cond}(Q_\perp^T \Lambda Q_\perp)}_{\gamma(Q,\Lambda)} \underbrace{\frac{\|Q_\perp^T \Lambda x\|}{\|Q_\perp^T \Lambda Q_\perp\|}}_{\epsilon(x,Q,\Lambda)} \quad . \tag{6.15}$$

Scaling Λ has no effect on the solution to (6.4), nor does it have any effect on these two terms.

Condition Term

The condition term $\gamma(Q, \Lambda) \geq 1$ is a unit-less gain factor that determines how the condition number of the system defined by the projection matrix Q and precision matrix Λ amplifies the other term $\epsilon(x, Q, \Lambda)$. Analogous to the way in which the condition number of a symmetric positive-definite matrix A is the ratio of its maximum to its minimum eigenvalue

$$\mathrm{cond}(A) = \left(\max_{x \neq 0} \frac{x^T A x}{x^T x} \right) \Big/ \left(\min_{x \neq 0} \frac{x^T A x}{x^T x} \right) \quad , \tag{6.16}$$

the condition term $\gamma(Q, \Lambda) \geq 1$ measures the condition number of the precision matrix Λ for vectors confined to the nullspace of Q (proof in Section B.2)

$$\gamma(Q, \Lambda) = \left(\max_{x \neq 0, Qx = 0} \frac{x^T \Lambda x}{x^T x} \right) \Big/ \left(\min_{x \neq 0, Qx = 0} \frac{x^T \Lambda x}{x^T x} \right) \quad . \tag{6.17}$$

When this ratio is large, it means that some directions in the nullspace of the projection matrix are penalised far less than others, and small deviations from the mode of the prior distribution may affect the solution drastically in these directions.

To make the upper bound small, the precision matrix Λ should be chosen such that the matrix $Q_\perp^T \Lambda Q_\perp$ is well-conditioned (has a small condition number). This is stricter than the constraint from Section 6.4.2 that the matrix

be invertible for the problem to have a unique solution, since any symmetric matrix with finite condition number is invertible.

The eigenvalues of $Q_\perp^T \Lambda Q_\perp$ are each a different convex combination of the eigenvalues of Λ (proof in Section B.3). Using this property, it is possible to obtain an upper bound on the condition number

$$\text{cond}(Q_\perp^T \Lambda Q_\perp) \leq \text{cond}(\Lambda) \tag{6.18}$$

since, as a convex combination, the eigenvalues of $Q_\perp^T \Lambda Q_\perp$ must be bounded by the extremal eigenvalues of Λ

$$\lambda_{\min}(\Lambda) \leq \lambda_i(Q_\perp^T \Lambda Q_\perp) \leq \lambda_{\max}(\Lambda) \ . \tag{6.19}$$

However this is of limited use as Λ typically has a nullspace that contains at least the static non-zero trajectory, and therefore its condition number is infinite.

If the eigenvalues of $Q_\perp^T \Lambda Q_\perp$ were obtained as random convex combinations of the eigenvalues of Λ, then it would be more likely for the condition number of $Q_\perp^T \Lambda Q_\perp$ to be large if numerous eigenvalues of Λ were very large or very small compared to many others. To increase the likelihood of a well-conditioned system of equations in this scenario, the number of pairs of vastly dissimilar eigenvalues $|\{(i,j) : \lambda_i(A) \leq \epsilon \lambda_j(A)\}|$ should be few.

Error Term

The error term $\epsilon(x, Q, \Lambda) \geq 0$ measures the component of the true trajectory x that has high cost according to the precision matrix Λ, projected into the nullspace of the projection matrix Q using $I - P_Q = Q_\perp Q_\perp^T$

$$\epsilon(x, Q, \Lambda) = \|(I - P_Q)\Lambda x\| \Big/ \left(\max_{x \neq 0, Qx = 0} \frac{x^T \Lambda x}{x^T x} \right) \ . \tag{6.20}$$

The denominator normalises for the norm of the precision matrix Λ for vectors confined to the nullspace of the projection matrix Q. When a trajectory subspace is used, and the precision matrix is the projector $\Lambda = I - P_\Theta$, the numerator of $\epsilon(x, Q, \Lambda)$ is

$$\|(I - P_Q)(I - P_\Theta)x\| \ , \tag{6.21}$$

measuring the unobserved component of the trajectory following projection on to the complementary subspace.

6.4.5 Ramifications for the Subspace Prior

The critical nature of choosing the subspace dimension k is reflected in the bound on reconstruction error. Using the expression in (6.13) from the derivation of the bound gives

$$v(x, Q, I - P_\Theta) = \left\| \left(Q_\perp^T \left(I - P_\Theta \right) Q_\perp \right)^{-1} \right\| \cdot \left\| Q_\perp^T \left(I - P_\Theta \right) x \right\| \ . \tag{6.22}$$

The operator norm of the inverse matrix is monotonically *increasing* in k (proof in Section B.4). It is infinite when $3k > 2\ell$ since the matrix is singular, as outlined earlier. The second norm is not monotonic in k, although it is bounded

$$\left\| Q_\perp^T \left(I - P_\Theta \right) x \right\| \leq \| \left(I - P_\Theta \right) x \| \tag{6.23}$$

since $\|Q_\perp^T\| = 1$, and this envelope is monotonically *decreasing* in k (proof in Section B.5).

This exposes the two conflicting forces in reconstruction using a subspace. The subspace dimension should be chosen large to enable accurate representation of the trajectory, but it should be chosen small to avoid a large condition number. The bound in (6.11) suggests a simple adaptive strategy that can be performed without knowledge of x: choose the largest k (by exhaustive

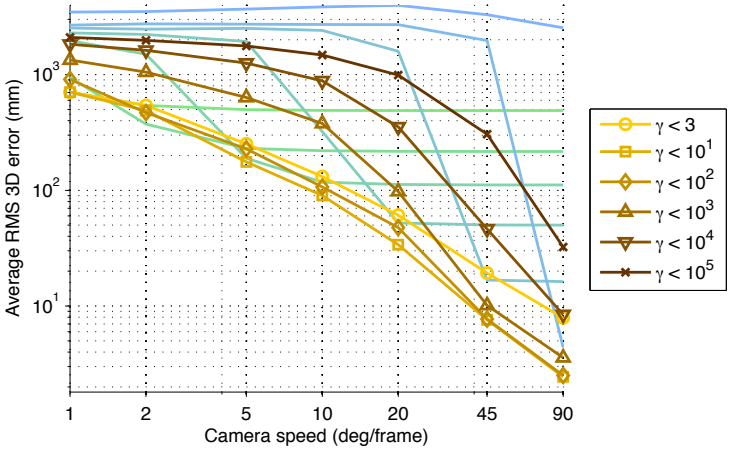


Figure 6.2: Reconstruction error versus orbiting camera speed using a DCT basis with size automatically determined using an upper limit on the condition term. The faint curves in the background are replicated from Figure 6.1.

search) such that $\gamma(Q, I - P_{\Theta\{1,\ldots,k\}}) \leq \gamma_{\max}$. Figure 6.2 shows that this is indeed an effective strategy. The efficacy of this approach is encouraging evidence that the bound is reasonably tight.

6.4.6 Bound for the Regularised Problem

A similar bound can be established for the unconstrained problem in (6.3), which trades off projection error against trajectory cost. The difference between the true trajectory x and the solution $x^\star$ is bounded

$$\begin{aligned}
\|x - x^\star\| &= \|x - (Q^T Q + \Lambda)^{-1} Q^T u\| \\
&= \|x - (Q^T Q + \Lambda)^{-1} Q^T Q x\| \\
&= \|(Q^T Q + \Lambda)^{-1} \Lambda x\| \\
&\leq \|(Q^T Q + \Lambda)^{-1}\| \cdot \|\Lambda x\| \\
&= \mathrm{cond}(Q^T Q + \Lambda) \frac{\|\Lambda x\|}{\|Q^T Q + \Lambda\|}
\end{aligned} \tag{6.24}$$

where $Q^TQ + \Lambda$ must be invertible for there to be a unique solution.

6.5 Toeplitz Precision Matrices from High-Pass Filters

The previous sections have shown that if the precision matrix is defined by a trajectory subspace and the subspace dimension k is chosen too large, then reconstruction may fail because the resulting system of equations is poorly conditioned. This section seeks alternative precision matrices that are less prone to this mode of failure. The precision matrix must be positive semidefinite and penalise unnatural motion. To minimise the risk of a poorly-conditioned system, as many of its eigenvalues as possible should be non-zero and similar in magnitude.

In the absence of any knowledge of the absolute time at which the sequence was captured, it might as well be assumed that the prior distribution is shift-invariant, and therefore that the trajectory is a stationary process with a Toeplitz covariance matrix. Whereas in object detection it was necessary to estimate the covariance matrix from data, this is not necessary in trajectory reconstruction because the properties of a likely trajectory are more easily intuited. In this direction, this work proposes that the distribution be specified directly in the form of a near-Toeplitz precision matrix defined by a high-pass filter. This is motivated by the observation that a stationary process that is defined on the set of all integers will have both a covariance matrix Σ and a precision matrix $\Lambda = \Sigma^{-1}$ (assuming that it exists) that are bi-infinite Toeplitz due to the result in Section 2.14. Furthermore, if Λ is a bi-infinite Toeplitz matrix that is symmetric and positive semidefinite, then there exists a unique bi-infinite Toeplitz operator $Gx = g * x$ such that $\Lambda = G^TG$ [24] and the objective would therefore correspond to the norm of

a convolution $x^T \Lambda x = \|g * x\|^2$.

Penalising the response of a trajectory to a high-pass filter will discourage high-frequency motion. If the point is assumed to have constant mass, then minimising the response to the first-difference filter $d_1 = (-1, 1)$ will seek the trajectory with the minimum average kinetic energy $\sum_t \frac{1}{2} m \|v[t]\|^2$, and minimising the response to the second-difference filter $d_2 = (-1, 2, -1)$ effectively imposes the assumption that the point is subject to random i.i.d. forces drawn from a Gaussian distribution [54].

The boundaries of the finite signal are handled by excluding the elements of the convolution where the support of the filter is not contained entirely within the trajectory. If a filter g has support m, then the Toeplitz matrix that computes convolution $Gx = g * x$ with a signal of length $\ell \geq m$ will have dimension $(\ell - m + 1) \times \ell$ and therefore a nullspace of dimension $m - 1$. The matrix corresponding to a first-difference filter is $(\ell - 1) \times \ell$

$$D_1 = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & -1 & \\ & & & & 1 & -1 \end{bmatrix} \tag{6.25}$$

and the matrix corresponding to a second-difference filter is $(\ell - 2) \times \ell$

$$D_2 = \begin{bmatrix} -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \end{bmatrix}. \tag{6.26}$$

The first-difference operator has the stationary trajectory $x = (1, 1, \ldots, 1)$ in

its nullspace and the second-difference operator has the stationary trajectory and the constant velocity trajectory $x = (1, 2, \ldots, \ell)$ in its nullspace.

The precision matrices defined by these filters are not exactly Toeplitz due to boundary effects. However, they do become Toeplitz just a few samples away from the boundaries. The precision matrix that corresponds to the first-difference operator is

$$D_1^T D_1 = \left[\begin{array}{cccccccc} 1 & -1 & & & & & & \\ \hline -1 & 2 & -1 & & & & & \\ & -1 & 2 & -1 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & -1 & 2 & -1 & & \\ & & & & -1 & 2 & -1 & \\ \hline & & & & & -1 & 1 \end{array}\right] \tag{6.27}$$

and the precision matrix defined by the second-difference operator is

$$D_2^T D_2 = \left[\begin{array}{cccccccccc} 1 & -2 & 1 & & & & & & & \\ -2 & 5 & -4 & 1 & & & & & & \\ \hline 1 & -4 & 6 & -4 & 1 & & & & & \\ & 1 & -4 & 6 & -4 & 1 & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & 1 & -4 & 6 & -4 & 1 & & \\ & & & & 1 & -4 & 6 & -4 & 1 & \\ \hline & & & & & 1 & -4 & 5 & -2 & \\ & & & & & & 1 & -2 & 1 & \end{array}\right]. \tag{6.28}$$

The use of compact filters not only minimises the boundary effects but also encodes conditional independence. Off-diagonal elements of the precision matrix are zero if and only if the two variables are conditionally independent [35]. This captures the intuition that the initial position of a point's

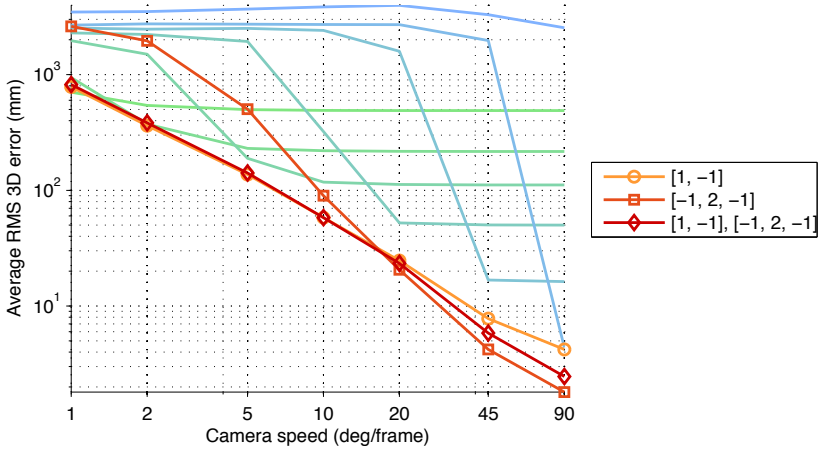


Figure 6.3: Reconstruction error versus orbiting camera speed for simple trajectory filters. Superior results are achieved over all sequences without having to tune any parameters.

trajectory should not be related to its final position directly, but only indirectly through a sequence of adjacent frames.

To construct a precision matrix for three-dimensional trajectories, the filters are simply applied independently in each dimension. For the particular vectorisation $x = (x_1[0], x_2[0], x_3[0], x_1[1], \ldots, x_3[\ell - 1])$, the precision matrix is formed

$$x^T \Lambda x = \sum_{p=1,2,3} \|g \star x_p\|^2 = \|(G \otimes I_3)x\|^2 = x^T (G^T G \otimes I_3) x \ . \tag{6.29}$$

If v is in null(G), then there are three corresponding vectors in null(Λ) since $Gv = 0$ implies $\Lambda(v \otimes I_3) = 0$.

The effectiveness of using a precision matrix constructed from convolution operators is evident in Figure 6.3. Simple filters achieve equal or better reconstruction error than the subspace approach. The experiments only consider the filters $(-1, 1)$ and $(-1, 2, -1)$ since these approximate with minimal support the first- and second-derivative respectively. The robustness of the

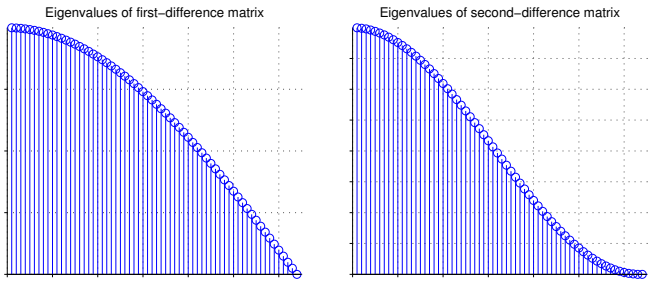


Figure 6.4: The eigenvalues of G^TG where Gx computes the convolution of x with the first-difference filter $(-1, 1)$ and second-difference filter $(-1, 2, -1)$. The first-difference filter has one eigenvalue that is zero, and the second-difference filter has two.

convolutional prior compared to the subspace prior can be explained by examining the eigenvalues of the precision matrix. The eigenvalues of the subspace projector $I - P_\Theta$ are all one or zero, whereas the eigenvalues of first- and second-difference filters, shown in Figure 6.4, vary smoothly and are mostly non-zero When using a trajectory subspace, catastrophic failure occurs when there exists one direction in null(Q) that is approximately in the subspace and one direction in null(Q) that is approximately orthogonal to it, since the condition number approaches infinity. Filters avoid the conditioning problem by having a continuous spectrum of eigenvalues with very few zeros. In fact, the convolutional precision matrix does still have a nullspace, although its dimension is greatly reduced compared to the subspace prior. A precision matrix may be formed from multiple filters by taking a positive linear combination of their individual precision matrices. The second-difference filter has many more near-zero eigenvalues than the first-difference filter, and therefore adding a small relative component (0.01) of the first-difference cost to the second-difference cost prevents a poorly-conditioned system from resulting at low camera velocities in Figure 6.3.

6.6 Implicit Stationarity in the DCT Subspace Constraint

The DFT is to periodic convolution as the Discrete Cosine and Sine Transforms (DCT and DST, collectively the Discrete Trigonometric Transforms, DTTs) are to *symmetric* periodic convolution. That is, the periodic convolution of two symmetric periodic sequences can be computed as element-wise multiplication using the DTTs. However, the symmetric case is greatly complicated by the many different classes of symmetric periodic signal. Each class defines a different transform, resulting in eight DCTs and eight DSTs. In this section it will be shown that minimising the distance from a DCT subspace is equivalent to minimising the response of the trajectory to a high-pass filter.

6.6.1 Symmetric Periodic Signals

To understand symmetric periodic convolution and the DTTs, it is important to be familiar with the different types of symmetric periodic signal.

A continuous signal x can be either symmetric $x(t) = x(-t)$ or anti-symmetric $x(t) = -x(-t)$. If x is symmetric or anti-symmetric $x(t) = sx(-t)$ with sign $s \in \{-1, 1\}$, as well as periodic $x(t) = x(t + 2n)$ with period $2n$, then it has the same type of symmetry at the midpoint $t = n$ since

$$x(n + t) = x(-n + t) = sx(n - t) \ . \tag{6.30}$$

Conversely, any function which has the same type of symmetry at $t = 0$ and $t = n$, such that $x(t) = sx(-t)$ and $x(n + t) = sx(n - t)$, is periodic with period $2n$ since

$$x(t + 2n) = x(n + (t + n)) = sx(n - (t + n)) = sx(-t) = x(t) \ . \tag{6.31}$$

(This is analogous to standing between two mirrors and observing an infinite symmetric periodic reflection.) Thus a symmetric periodic signal is sufficiently described in the values that it takes on the interval $[0, n]$ and the type of symmetry that occurs at the two boundaries.

Symmetry is more complicated in discrete signals since, in addition to being symmetric or anti-symmetric, reflections can occur on or between samples:

$$x[\tau + t] = \pm\, x[\tau - t], \quad \text{or} \quad x[\tau + t] = \pm\, x[\tau - 1 - t] \ . \tag{6.32}$$

The former is known as whole-sample symmetry and the latter as half-sample. For example, the sequence C, B, A, B, C has whole-sample symmetry, and the sequence C, B, A, A, B, C has half-sample symmetry. This gives four types of discrete symmetry, which Martucci [44] denotes

$$\{\mathrm{WS}, \mathrm{WA}, \mathrm{HS}, \mathrm{HA}\} = \{\mathrm{W}, \mathrm{H}\} \times \{\mathrm{S}, \mathrm{A}\}$$

for Whole/Half and Symmetric/Anti-symmetric. Note that symmetry of type WA at $t = \tau$ implies that $x[\tau] = -x[\tau] = 0$.

The type of a symmetric periodic signal is determined by the symmetry on each side. This gives a total of $4^2 = 16$ types of symmetric periodic signal, which Martucci denotes

$$\{\mathrm{WSWS}, \mathrm{WSWA}, \dots\} = (\{\mathrm{W}, \mathrm{H}\} \times \{\mathrm{S}, \mathrm{A}\})^2 \ .$$

Each DTT maps from one type of symmetric periodic signal in the original domain to a symmetric periodic signal in the transform domain, often of a different type. This is listed for the relevant transforms in Table 6.1.

The sixteen transforms are typically identified as an element of the set

$$\{\mathrm{DCT}, \mathrm{DST}\} \times \{1, 2, 3, 4\} \times \{\mathrm{E}, \mathrm{O}\} \ .$$

T_a	Name	a	$\bar{a}$	Defining Elements
C_1	DCT-1	WSWS	WSWS	$n+1$
S_1	DST-1	WAWA	WAWA	$n-1$
C_2	DCT-2	HSHS	WSWA	n
S_2	DST-2	HAHA	WAWS	n

Table 6.1: Transform T_a maps signals of type a in the original domain to signals of type $\bar{a}$ in the frequency domain. Both signals have period $2n$ and are defined by the same number of elements.

Cosine transforms are symmetric at $t = 0$ and sine transforms are anti-symmetric. The letters E and O represent even and odd transforms. Odd transforms arise when whole-sample symmetry occurs on one side and half-sample symmetry on the other. It's difficult to imagine a situation where this would be desirable, and all transforms are henceforth assumed to be even. If the two types of symmetry differ in that one is symmetric and one is anti-symmetric, then the signal is not periodic but anti-periodic $x[t+T] = -x[t+T]$. This corresponds to transforms of type 3 and 4. Anti-periodic signals can only be convolved with anti-periodic signals, since the convolution of a periodic and an anti-periodic signal is a trivial signal of all zeros. Since anti-periodic signals must possess an anti-symmetric boundary, and it does not make sense to consider the anti-symmetric extension of a point trajectory, these transforms will not be considered. The only transforms which remain are the DCT-1, DST-1, DCT-2 and DST-2.

6.6.2 Symmetric Periodic Convolution

If u and v are symmetric periodic signals of compatible types, then their circular convolution is a symmetric periodic signal up to a possible one-

$u * v = L_\tau w$				$T_a u \odot T_b v = s T_c w$						
a	b	c	τ	T_a	T_b	T_c	s	$\bar{a}$	$\bar{b}$	$\bar{c}$
WSWS	WSWS	WSWS	0	C_1	C_1	C_1	1	WSWS	WSWS	WSWS
WSWS	WAWA	WAWA	0	C_1	S_1	S_1	1	WSWS	WAWA	WAWA
WSWS	HSHS	HSHS	0	C_1	C_2	C_2	1	WSWS	WSWA	WSWA
WSWS	HAHA	HAHA	0	C_1	S_2	S_2	1	WSWS	WAWS	WAWS
WAWA	WAWA	WSWS	0	S_1	S_1	C_1	-1	WAWA	WAWA	WSWS
WAWA	HSHS	HAHA	0	S_1	C_2	S_2	1	WAWA	WSWA	WAWS
WAWA	HAHA	HSHS	0	S_1	S_2	C_2	-1	WAWA	WAWS	WSWA
HSHS	HSHS	WSWS	-1	C_2	C_2	C_1	1	WSWA	WSWA	WSWS
HSHS	HAHA	WAWA	-1	C_2	S_2	S_1	1	WSWA	WAWS	WAWA
HAHA	HAHA	WSWS	-1	S_2	S_2	C_1	-1	WAWS	WAWS	WSWS

Table 6.2: The ten convolution properties for signal types in {WSWS, WAWA, HSHS, HAHA}. Convolution of two signals in the original domain is equivalent to element-wise multiplication of the signals in the transform domain. The signal types of u, v and w are a, b and c, and the types of $T_a u$, $T_b v$ and $T_c w$ are $\bar{a}$, $\bar{b}$ and $\bar{c}$.

sample shift

$$u * v = L_\tau w \ . \tag{6.33}$$

Here w is a symmetric periodic signal, L_τ is a translation operator $(L_\tau x)[t] = x[t+\tau]$ and $\tau \in \{0, -1\}$. The signal types of u and v need not be the same, and the type of w is determined by the types of the input signals, as defined in Table 6.2.

For each possible convolution of two symmetric periodic signals, there is

Type	Defining Elements	Number
WSWS	$t = 0, \ldots, n$	$n + 1$
WAWA	$t = 1, \ldots, n - 1$	$n - 1$
HSHS	$t = 0, \ldots, n - 1$	n
HAHA	$t = 0, \ldots, n - 1$	n
WSWA	$t = 0, \ldots, n - 1$	n
WAWS	$t = 1, \ldots, n$	n
HSHA	$t = 0, \ldots, n - 1$	n
HAHS	$t = 0, \ldots, n - 1$	n

Table 6.3: The set of sufficient samples for signals of period $2n$ varies for different types of symmetric periodic signal.

a corresponding multiplication in the transform domain

$$T_a u \odot T_b v = s T_c w \tag{6.34}$$

where $s \in \{-1, 1\}$ and the transforms T_a, T_b and T_c are determined by the types of the signals u and v. Whereas the DFT has a single convolution property, the DTTs have forty distinct convolution properties for different combinations of signal types. The relevant subset of convolution properties is listed in Table 6.2.

Up to this point, the DTTs have been considered maps from and to infinite signals. Any symmetric periodic signal can be represented by a finite vector of defining elements. The finite versions of the DTTs are obtained by considering a map from the minimal defining elements in the original domain to the minimal defining elements in the transform domain. Finite DTTs are more complicated than DFTs since the minimal set of defining elements depends on the type of symmetry, as indicated in Table 6.3.

Let $[T]$ denote the finite version of transform T. To derive the finite transform, introduce invertible extension operators E_a which map from a finite signal to a symmetric periodic sequence of type a, for example

$$(E_{\text{wsws}}x)[t] = \begin{cases} x[t], & 0 \leq (t \bmod 2n) < n \\ x[2n-t], & n \leq (t \bmod 2n) < 2n \end{cases} \tag{6.35}$$

where $x[t]$ is only defined on $0 \leq t < n+1$. Using the fact that the transform T_a of a signal of type a is of type $\bar{a}$, the finite transform is defined such that

$$E_{\bar{a}}[T_a]x = T_a E_a x \quad \forall x \tag{6.36}$$

and therefore $[T_a] = E_{\bar{a}}^{-1} T_a E_a$.

The finite transforms that preserve the convolution property are not orthogonal, however they are related to an orthogonal transform, denoted $[\tilde{T}_a]$, by a diagonal transformation on either side $[\tilde{T}_a] = \text{diag}(\alpha_a)[T_a]\,\text{diag}(\beta_a)$ [44]. The orthogonal transform satisfies $[\tilde{T}_a]^T[\tilde{T}_a] = [\tilde{T}_a][\tilde{T}_a]^T \propto I$.

6.6.3 Equivalent Filter

Theorem 6.2. *There exists a symmetric periodic signal $E_{\text{wsws}}y$ such that the norm of (one symmetric half-period of) its circular convolution with the symmetric periodic extension $E_{\text{hshs}}x$ of a finite signal $x : \{0, \ldots, \ell-1\} \to \mathbb{R}$ is equal to the component of that signal that is orthogonal to the $k \geq 1$ lowest frequencies of the finite (orthogonal) DCT-2 basis*

$$\|z\|^2 = \|(I - \Phi\Phi^T)x\|^2 \tag{6.37}$$

where $z : \{0, \ldots, \ell-1\} \to \mathbb{R}$ satisfies

$$E_{\text{hshs}}z = E_{\text{hshs}}x * E_{\text{wsws}}y \tag{6.38}$$

and Φ is the $\ell \times k$ matrix comprising the first k columns of the orthogonal DCT-2 transform $[\Phi \mid \Phi_\perp] = [\tilde{C}_2]^T$.

Proof. The subspace penalty can be expressed

$$\|(I - \Phi\Phi^T)x\|^2 = \|\Phi_\perp^T x\|^2 = \|\operatorname{diag}(h)[\tilde{C}_2]x\|^2 \tag{6.39}$$

where h selects only the high-frequency components

$$h[t] = \begin{cases} 0, & \text{if } 0 \le t < k \\ 1, & \text{if } k \le t < \ell \ . \end{cases} \tag{6.40}$$

The diagonal transforms that relate the transform $[C_2]$ to its orthogonal form $[\tilde{C}_2] = \operatorname{diag}(\alpha_{\text{HSHS}})[C_2]\operatorname{diag}(\beta_{\text{HSHS}})$ are [44]

$$\alpha_{\text{HSHS}}[t] = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } t = 0 \\ 1, & \text{if } 1 \le t < \ell \end{cases} \tag{6.41}$$

and $\beta_{\text{HSHS}} = 1$. Therefore the subspace penalty is simply

$$\|(I - \Phi\Phi^T)x\|^2 = \|\operatorname{diag}(h)[C_2]x\|^2 \tag{6.42}$$

since $h \odot \alpha_{\text{HSHS}} = h$.

The symmetric periodic extensions $E_{\text{HSHS}}x$, $E_{\text{WSWS}}y$ and $E_{\text{HSHS}}z$ all have period 2ℓ, which implies that the domain of y is $\{0, \ldots, \ell\}$ in accordance with Table 6.3. Using Table 6.2, the convolution

$$E_{\text{HSHS}}z = E_{\text{HSHS}}x * E_{\text{WSWS}}y \tag{6.43}$$

is equivalent to the multiplication

$$C_2 E_{\text{HSHS}}z = (C_2 E_{\text{HSHS}}x) \odot (C_1 E_{\text{WSWS}}y) \ . \tag{6.44}$$

This has an equivalent representation in terms of the extensions of the finite transforms $\hat{x} = [C_2]x$, $\hat{y} = [C_1]y$ and $\hat{z} = [C_2]z$

$$E_{\text{WSWA}}\hat{z} = (E_{\text{WSWA}}\hat{x}) \odot (E_{\text{WSWS}}\hat{y}) \tag{6.45}$$

where $\hat{x}$ and $\hat{z}$ have domain $\{0, \ldots, n-1\}$ and $\hat{y}$ has domain $\{0, \ldots, n\}$. Due to symmetry, this is satisfied if and only if

$$(E_{\text{WSWA}}\hat{z})[t] = (E_{\text{WSWA}}\hat{x})[t] \cdot (E_{\text{WSWS}}\hat{y})[t] \qquad t = 0, \ldots, \ell \ . \tag{6.46}$$

However, since $(E_{\text{WSWA}}\hat{x})[\ell] = 0$ and $(E_{\text{WSWA}}\hat{z})[\ell] = 0$ due to WSWA extension, this is equivalent to

$$\hat{z}[t] = \hat{x}[t] \cdot \hat{y}[t] \qquad t = 0, \ldots, \ell - 1 \tag{6.47}$$

with $\hat{y}[\ell]$ arbitrary. This can also be written $\hat{z} = \operatorname{diag}(P\hat{y})\hat{x}$ where P is the operator that selects elements $\{0, \ldots, \ell - 1\}$ from a signal with domain $\{0, \ldots, \ell\}$. The norm of (half of the symmetric period of) the convolution is

$$\|z\|^2 = \|[\tilde{C}_2]z\|^2 = \|\operatorname{diag}(\alpha_{\text{HSHS}})[C_2]z\|^2 = \|\operatorname{diag}(\alpha_{\text{HSHS}} \odot P\hat{y})[C_2]x\|^2 \ . \tag{6.48}$$

Therefore the two expressions are equal if and only if $y = [C]^{-1}\hat{y}$ where $\hat{y}$ satisfies $P\hat{y} = h$. □

The filter $y = [C]^{-1}\hat{y}$ whose transform satisfies $P\hat{y} = h$ and $\hat{y}[\ell] = 0$ is shown in the top of Figure 6.5.

6.7 Alternative Forms of Trajectory Prior

Salzmann and Urtasun [54] considered trajectory reconstruction where additional information about the sequence is encoded in the choice of prior. To reconstruct a sequence containing collisions, they incorporate group-L_1 regularisation of the acceleration to encourage impulse forces that are sparse in time. To reconstruct the parabolic motion of a projectile, they inject a constant unknown gravitational acceleration. This provides a mechanism to improve the quality of the reconstruction through manual intervention.

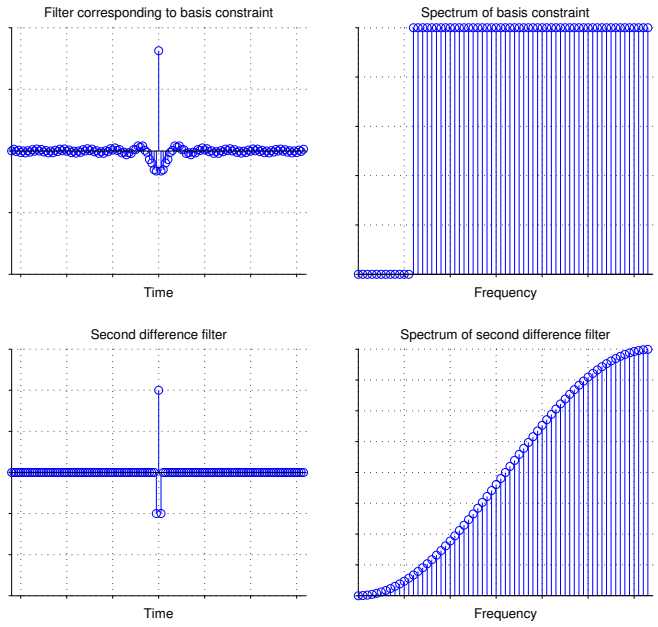


Figure 6.5: Projection on to the high frequency bases of the DCT-2 is equivalent to (symmetric periodic) convolution with a high-pass filter (top left) whose transform is a high-pass step function (top right). The second-difference filter and its transform are shown for comparison (bottom).

Zhu and Lucey [71] formulate trajectory reconstruction as compressed sensing using a convolutional dictionary of trajectories that is learnt offline. They use the bound developed in this work to motivate compressed sensing as a method to circumvent the limit imposed by the condition number.

6.8 Reconstruction of Real Image Sequences

The convolutional precision matrix was qualitatively compared to the DCT subspace constraint of Park et al. [50] in a number of real sequences. Representative reconstructions are presented in Figures 6.7, 6.9 and 6.11. It is typically observed that, while the solution using a DCT subspace constraint is smoother, the convolutional precision matrix produces a more realistic trajectory. For example, note the triangular path of the feet and the more complex path of the swinging arm in Figure 6.7. More importantly, however, the convolutional precision matrix did not require any parameter selection. To obtain a reasonable reconstruction in non-synthetic sequences, it is generally necessary to use multiple cameras to simulate a single fast-moving camera, and to even have multiple simultaneous views in some frames.

Figure 6.6: Every second frame of the “dance” sequence.

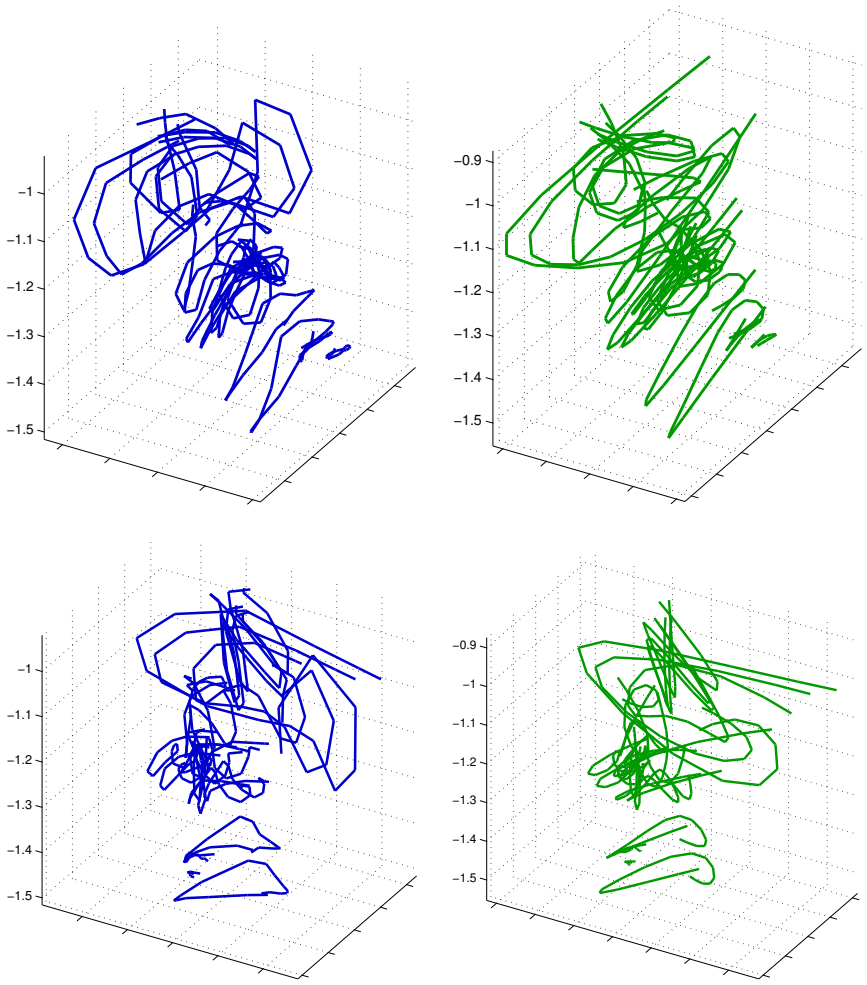


Figure 6.7: Trajectory filtering (left) and a truncated DCT basis (right) achieve similarly plausible reconstruction on real-world examples where no ground-truth is available. The filter-based reconstruction used the sum of the response of $(-1, 1)$ and $(-1, 2, -1)$ filters and did not require any hand-tuning. A $k = 6$ dimensional DCT basis had to be chosen by trial and error.

Figure 6.8: Every third frame of the "hand wave" sequence.

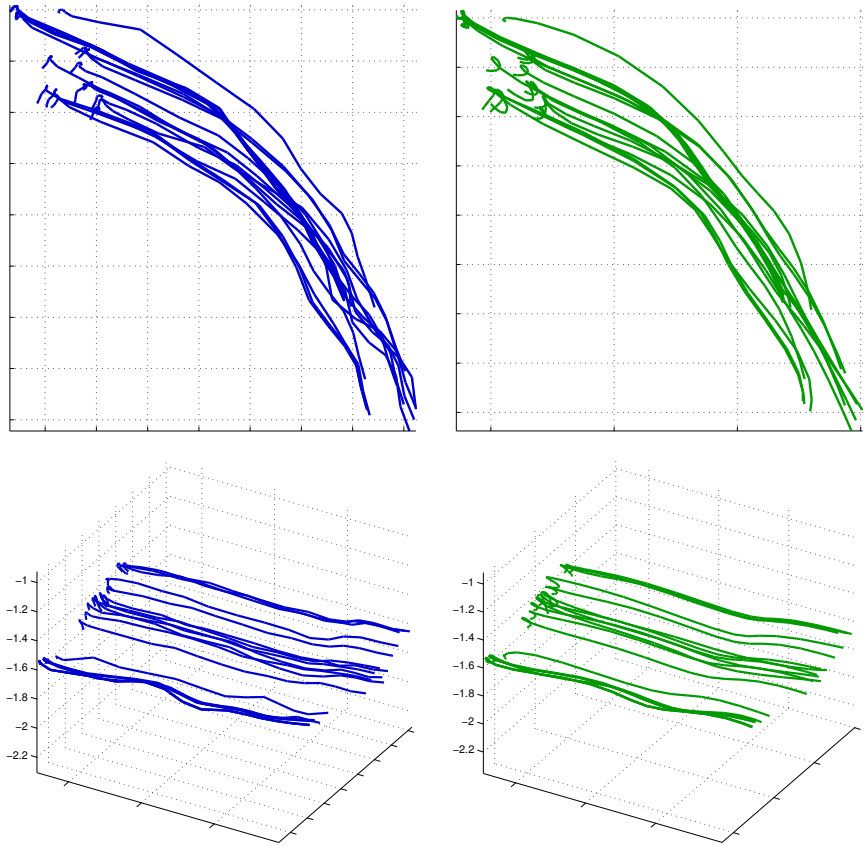


Figure 6.9: Reconstruction of the "hand wave" sequence using filters (left) and DCT (right).

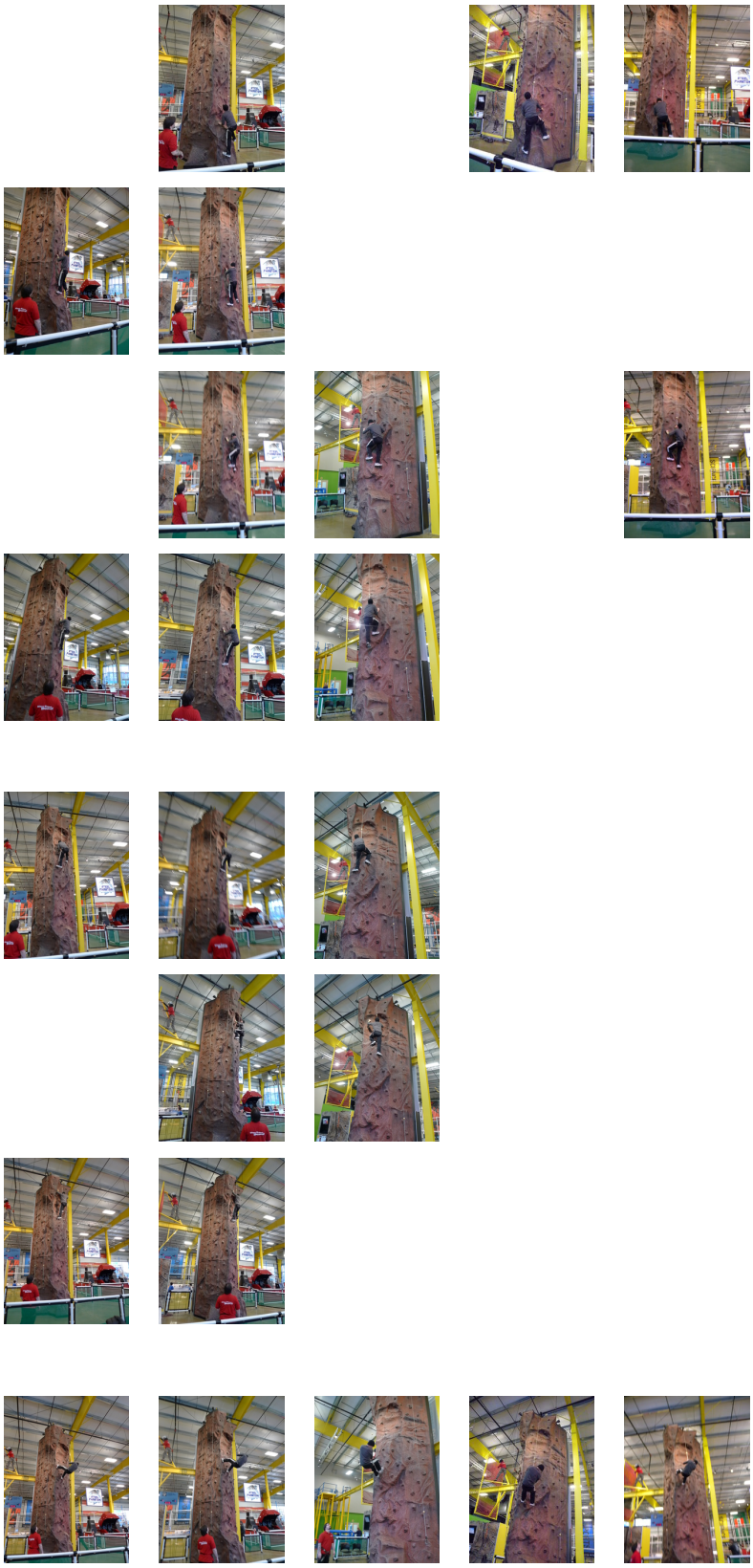


Figure 6.10: Every fourth frame of the "rock climbing" sequence.

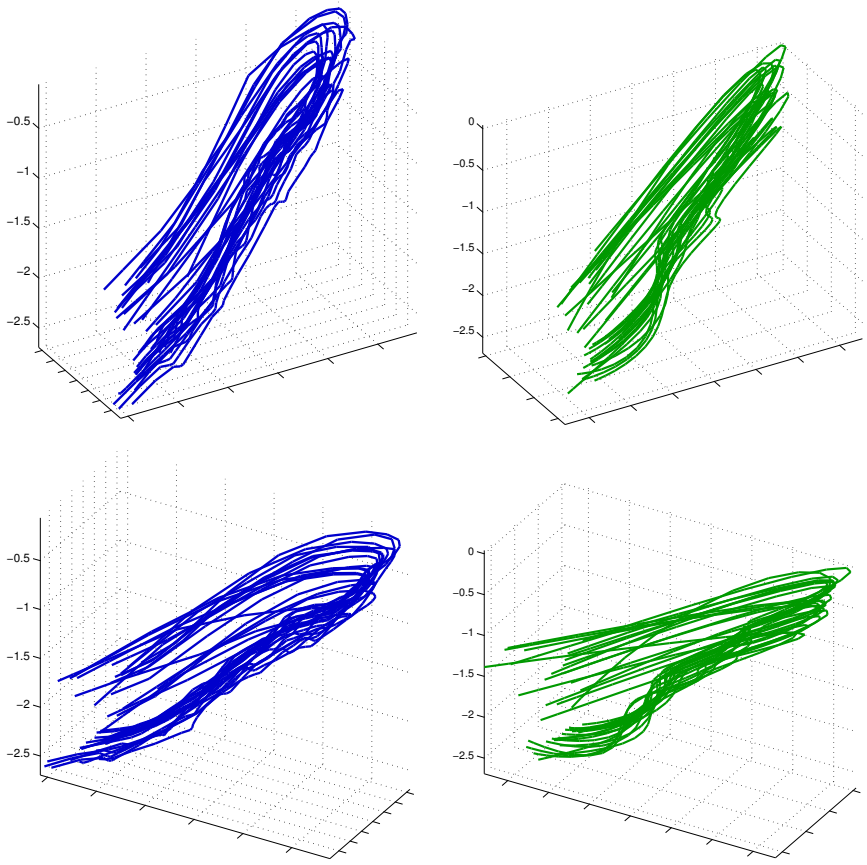


Figure 6.11: Reconstruction of the "rock climbing" sequence using filters (left) and DCT (right).

6.9 Combinatorial Trajectory Reconstruction

6.9.1 Overview

This chapter has so far considered the most general trajectory reconstruction problem where the position of a point may be any real-valued vector. *Combinatorial* trajectory reconstruction instead considers the scenario where the trajectory is confined to a finite but exponential set

$$\arg\min_{x} \; f(x) \quad \text{s.t.} \quad x[t] \in \mathcal{X}_t \quad t = 0, \ldots, \ell - 1 \;\; . \tag{6.49}$$

If the number of possible positions in every frame is $s = |\mathcal{X}_t|$, then there are s^ℓ combinations that define a possible trajectory. It is prohibitively expensive to exhaustively evaluate the objective for every trajectory because the number of trajectories is exponential in the length ℓ of the sequence: with $s = 2$ every additional frame doubles the running time.

Combinatorial trajectory reconstruction has previously been considered by Park and Sheikh [49] in the context of articulated motion. They proposed to search for the trajectory with the minimum component orthogonal to a subspace $f(x) = \|(I - \Theta\Theta^T)x\|^2$ using Branch and Bound. Each constraint set $\mathcal{X}_t$ can be relaxed to its convex hull so that a lower bound is obtained by solving a convex quadratic program. However, the worst-case running time of Branch and Bound is still exponential [47] in the length of the sequence. This work recognises that the objective defined by compact filters can be solved in time that is *linear* in the sequence length.

6.9.2 Graphical Model Interpretation

The problem of identifying the trajectory in the finite set that minimises the objective function can be understood as Maximum A Posteriori (MAP)

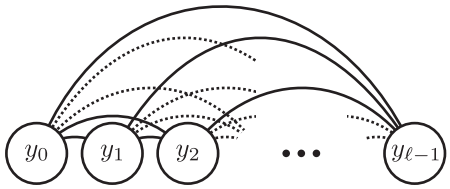


Figure 6.12: Minimising an objective function that projects trajectories on to a DCT subspace corresponds to inference in a fully connected graphical model since the maximal clique must contain all variables to compute dense inner products. This can only be solved exhaustively.

inference in an undirected graphical model. The joint distribution of the model is defined by a sum over the maximal cliques $\mathcal{C}$ of a graph [47]

$$-\ln p(x) = \sum_{C \in \mathcal{C}} f_C(x_C) \ . \tag{6.50}$$

The graph defined by an objective function contains a vertex for each variable and an edge for every two variables that appear together in a term. If the graph is a tree, then the min-sum algorithm can be applied to solve MAP inference in $O(\ell s^2)$ time [47].

DCT Subspace

The DCT subspace objective is not amenable to optimisation by the min-sum algorithm since its graph is complete (fully connected) as depicted in Figure 6.12. This structure arises because every term in the objective depends on every point in the trajectory through a dense inner product

$$f(x) = \|\Theta_\perp^T x\|^2 = \sum_{i=k+1}^{\ell} \sum_{p=1}^{3} \left(\theta_{(i,p)}^T x\right)^2 = \sum_{i=k+1}^{\ell} \left\| \sum_{t=0}^{\ell-1} \phi_i[t]\, x[t] \right\|^2 \ . \tag{6.51}$$

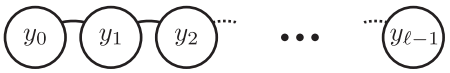


Figure 6.13: The graph associated with a first-difference objective is a tree and therefore inference can be performed in time that is linear instead of exponential in the length of the sequence.

First-Difference Filter

If the objective instead measures the response of the trajectory to a first-difference filter

$$f(x) = \sum_{t=0}^{\ell-2} \|x[t] - x[t+1]\|^2 \tag{6.52}$$

then the associated graph is a tree as shown in Figure 6.13. Therefore the min-sum algorithm can be applied to obtain the minimiser in $O(\ell s^2)$ time, reducing the complexity from exponential to linear in the sequence length. However, as will be shown in the following section, a first-difference filter is often insufficient.

Filter with Compact Support

The graph defined by a second-difference filter, depicted in Figure 6.14, is not a tree because it contains cycles. Luckily, the min-sum algorithm can be generalised [22] to objectives that comprise terms of m consecutive elements

$$f(x) = \sum_{t=0}^{\ell-m} h_t(x[t], x[t+1], \ldots, x[t+m-1]) \ . \tag{6.53}$$

The corresponding graph is said to have treewidth $m-1$ (trees have treewidth 1) and the minimiser can be found using the min-sum algorithm in $O(\ell s^m)$ time [22, 47]. This class of functions includes those that measure the response

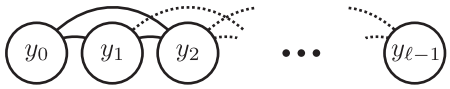


Figure 6.14: The graph that corresponds to a second-difference objective is not a tree. However, it is still amenable to efficient optimisation using the min-sum algorithm because it has small treewidth.

of the trajectory to a filter g that has support m

$$f(x) = \sum_{p=1}^{3} \|g \star x_p\|^2 = \sum_{t=0}^{\ell-m} \left\| \sum_{\tau=0}^{m-1} g[\tau]\, x[t+\tau] \right\|^2 . \tag{6.54}$$

Therefore the second-difference objective can be minimised in $O(\ell s^3)$ time.

6.10 Application: Articulated Trajectory Reconstruction

Articulated motion is a special case of non-rigid motion. Whereas in rigid motion, the distance between *every* pair of points remains constant, in articulated motion, the distance between points is preserved for only a subset of pairs, identified by the edges of a graph.

6.10.1 Formulation

The anatomy of an articulated body with n points is described by an undirected graph with vertices $\mathcal{V} = \{1, \ldots, n\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Let the configuration of the body at time t be represented $x_i[t] \in \mathbb{R}^3$ for points $i \in \mathcal{V}$. Each edge $(i, j) \in \mathcal{E}$ has a length $d_{ij} \geq 0$ which defines the 3D distance between points i and j

$$\|x_i[t] - x_j[t]\| = d_{ij} \tag{6.55}$$

for all t. It will be assumed that the graph is a tree: there exists a path between every two vertices, and the graph contains no cycles.

Each point i is observed at time t as projection $w_i[t] = P_t(x_i[t], \xi_t)$. This is equivalent to the linear constraint

$$Q_{ti} x_i[t] = u_{ti} \tag{6.56}$$

as in (5.9). The problem is thus to find the minimum cost trajectory that satisfies (6.55) and (6.56) given cameras, 2D projections, the anatomy of the articulated body and the precision matrix Λ of a Gaussian trajectory distribution

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{n} \|x_i\|_{\Lambda}^2 \\ \text{subject to} \quad & Q_{ti} x_i[t] = u_{ti} \qquad && i = 1, \ldots, n \\ & && t = 0, \ldots, \ell - 1 \\ & \|x_i[t] - x_j[t]\| = d_{ij} && (i,j) \in \mathcal{E} \\ & && t = 0, \ldots, \ell - 1 \ . \end{aligned} \tag{6.57}$$

6.10.2 Finite Feasible Set

Park and Sheikh [49] recognised that if the position of a point in one frame is known, then there are at most two feasible solutions for the positions of each of its neighbours in that frame. The binary ambiguity results from the intersection of the projection ray of (6.55) with the articulation sphere of (6.56), illustrated in Figure 6.15.

Assume that the position $x_i[t]$ is known. Since each Q_{tj} has a one-dimensional nullspace, the set of points that satisfy the projection constraint for $x_j[t]$ can be parameterised

$$\{x \in \mathbb{R}^3 : Q_{tj} x = u_{tj}\} = \{Q_{tj}^{\dagger} u_{tj} + z v_{tj} : z \in \mathbb{R}\} \tag{6.58}$$

where $Q_{tj}^{\dagger}$ denotes the pseudo-inverse and $v_{tj} \in \mathbb{R}^3$ is a unit vector $\|v_{tj}\| = 1$ in the nullspace $Q_{tj} v_{tj} = 0$. Substituting this form into the articulation

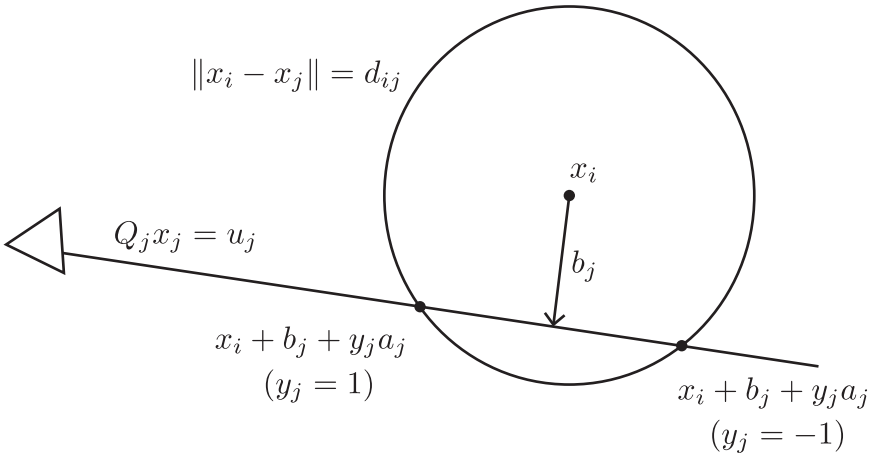


Figure 6.15: The intersection of the projection constraint and the articulation constraint has at most two solutions per frame, parameterised by a binary variable $y_j[t] \in \{-1, 1\}$. This holds for perspective and affine cameras. Time indices t are excluded from the figure for clarity.

constraint (6.55) yields a quadratic equation in the scalar z

$$\|Q_{tj}^{\dagger} u_{tj} + z v_{tj} - x_i[t]\|^2 = d_{ij}^2 \tag{6.59}$$

that has at most two solutions. The precise form of these solutions can be found by decomposing the norm into orthogonal components

$$\|P_{Q_{tj}}(Q_{tj}^{\dagger} u_{tj} - x_i[t])\|^2 + \|(I - P_{Q_{tj}})(z v_{tj} - x_i[t])\|^2 = d_{ij}^2 \tag{6.60}$$

where $P_{Q_{tj}} = Q_{tj}^{\dagger} Q_{tj}$ is the 3×3 projector on to the two-dimensional row-space of the projection matrix. This in turn gives

$$(z - v_{tj}^T x_i[t])^2 = d_{ij}^2 - \|Q_{tj}^{\dagger}(u_{tj} - Q_{tj} x_i[t])\|^2 \tag{6.61}$$

since $I - P_{Q_{tj}} = v_{tj} v_{tj}^T$. Thus the two solutions are enumerated by a binary variable $y_j[t] \in \{-1, 1\}$

$$x_j[t] = x_i[t] + a_j[t]\, y_j[t] + b_j[t] \tag{6.62}$$

where

$$a_j[t] = v_{tj}\sqrt{d_{ij}^2 - \|Q_{tj}^{\dagger}(u_{tj} - Q_{tj}x_i[t])\|^2} \ , \tag{6.63}$$

$$b_j[t] = Q_{tj}^{\dagger}(u_{tj} - Q_{tj}x_i[t]) \ . \tag{6.64}$$

If the camera is assumed to be orthographic, then the equation $Q_{ti}x_i[t] = u_{ti}$ is simply $R_t x_i[t] = w_i[t]$, and the parameterised trajectory has a simpler expression

$$a_j[t] = v_t\sqrt{d_{ij}^2 - \|w_j[t] - w_i[t]\|^2} \ , \tag{6.65}$$

$$b_j[t] = R_t^T(w_j[t] - w_i[t]) \tag{6.66}$$

where $R_t v_t = 0$.

6.10.3 Greedy Reconstruction of an Articulated Tree

Following the approach of Park and Sheikh [49], an algorithm for combinatorial trajectory reconstruction will be applied greedily to reconstruct the motion of an articulated tree by independently estimating the trajectory of each child node given that of its parent. The root node might be fixed to a rigid background, or its trajectory could be estimated using the methods for general reconstruction outlined earlier in the chapter. The greedy strategy is sub-optimal, but may still produce good reconstructions if the motion of all points is smooth.

The sub-problem of finding the trajectory of node j given the trajectory of its neighbour i

$$\begin{aligned} \min_{x_j} \quad & \|x_j\|_{\Lambda}^2 \\ \text{subject to} \quad & Q_{tj}x_j[t] = u_{tj} \qquad t = 0, \ldots, \ell - 1 \\ & \|x_i[t] - x_j[t]\| = d_{ij} \quad t = 0, \ldots, \ell - 1 \end{aligned} \tag{6.67}$$

is a combinatorial problem of the form in (6.49) where $f(x) = \|x\|_\Lambda^2$ and $\mathcal{X}_t = \{x_i[t] + b_j[t] + y \cdot a_j[t] : y \in \mathcal{Y}\}$ with $\mathcal{Y} = \{-1, 1\}$. Adopting a second-difference objective, this problem can be solved in $O(\ell s^m)$ time with $s = 2$ and $m = 3$ using the min-sum algorithm.

6.10.4 First-Difference Filters are Insufficient

Consider the reconstruction of an articulated trajectory whose parent node is stationary from the observations of a static orthographic camera $R_t = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right]$. The feasible positions in each frame are parameterised by $y[t] \in \mathcal{Y}$ according to

$$x_j[t] = \begin{bmatrix} \alpha[t] \\ \beta[t] \\ y[t]\,\gamma[t]) \end{bmatrix} \tag{6.68}$$

with $\gamma[t] \geq 0$. The trajectory that minimises the first-difference objective will satisfy $y[t] = y[t+1] = \zeta$ for all t since

$$\left\| \begin{bmatrix} \alpha[t] \\ \beta[t] \\ \zeta\,\gamma[t] \end{bmatrix} - \begin{bmatrix} \alpha[t+1] \\ \beta[t+1] \\ \zeta\,\gamma[t+1] \end{bmatrix} \right\|^2 \leq \left\| \begin{bmatrix} \alpha[t] \\ \beta[t] \\ \zeta\,\gamma[t] \end{bmatrix} - \begin{bmatrix} \alpha[t+1] \\ \beta[t+1] \\ -\zeta\,\gamma[t+1] \end{bmatrix} \right\|^2 \tag{6.69}$$

using the fact that $|\gamma[t] - \gamma[t+1]| \leq |\gamma[t] + \gamma[t+1]|$. Therefore, in this scenario, the optimal trajectory will never cross the z-plane, preferring instead to "bounce" off it to preserve the sign of the z component. This is particularly problematic for the reconstruction of real sequences with unknown parameters in Section 6.10.6. In contrast, the second-difference objective encourages trajectories to be smooth rather than slow, and is not susceptible to this issue.

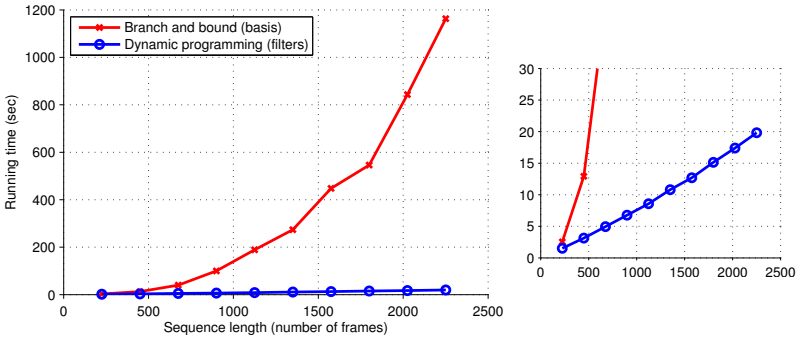


Figure 6.16: Running time versus sequence length for the two reconstruction algorithms. The trajectory basis objective may still take exponential time to solve despite the Branch and Bound algorithm (*left*). The dynamic programming method guarantees a solution in linear time (*right*). Both are globally optimal. The experiment is for an 18-joint human body sequence from CMU MoCap.

6.10.5 Experiment: Accuracy and Speed

The performance of the proposed algorithm was compared to an implementation of the Branch and Bound method using the same tools as Park and Sheikh [49] on simulated projections of sequences from the freely available CMU MoCap dataset (`http://mocap.cs.cmu.edu/`). In these experiments the ground truth perspective camera (constant throughout the sequence) and root node trajectory were supplied to the algorithm.

Figure 6.16 shows conclusively that dynamic programming is orders of magnitude more efficient for long sequences. In fact, despite employing a Branch and Bound strategy, the time complexity of the competing method still appears to grow exponentially. Both implementations were written in Matlab and neither is highly optimised. The fact that the running times are very similar for short sequences suggests that this is a fair comparison.

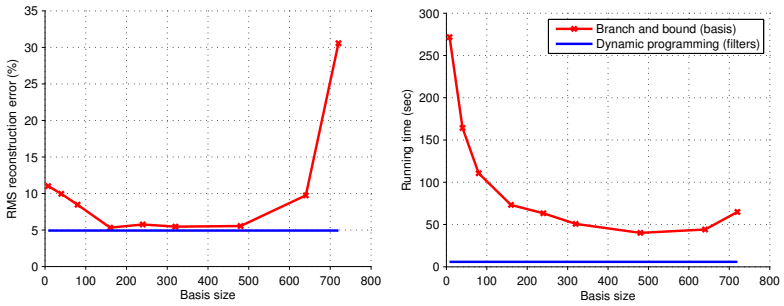


Figure 6.17: While the basis size in [49] is not critical, an incorrect choice still adversely affects the reconstruction (left). The filter objective obtains a slightly better reconstruction than the best basis reconstruction. The running time of the Branch and Bound method depends on the subspace dimension (right). Results were averaged over 8×800-frame sequences.

Another advantage of the filter-based approach is that there is no need to specify a subspace dimension. While Park and Sheikh [49] identified that articulated trajectory reconstruction is relatively insensitive to the number of DCT bases used, Figure 6.17 shows that this number does at least need to be chosen relatively large, and failure to do this correctly will still result in a poor reconstruction. It is also highlights that a sub-optimal choice affects the running time.

6.10.6 Reconstruction with Unknown Parameters

Articulated trajectory reconstruction typically assumes that the camera and root node positions can be recovered from the background using rigid Structure-from-Motion. For many "real-world" video sequences of interest, however, the background may lack sufficient structure or visual texture to reliably estimate cameras in this manner. Since articulated trajectories are relatively immune to the issue of reconstructability that arises from insufficient camera mo-

tion, it may be practical to assume a constant camera and reconstruct the relative motion of the structure within the camera reference frame. Small non-smooth camera motion (jitter) can be removed using 2D stabilisation.

To use a full-perspective camera would still require an estimate of the root trajectory in the camera reference frame. Adopting a weak-perspective camera model obviates this difficulty, and only requires the estimation of a scale parameter per frame α_t

$$Q_{ti} = \begin{bmatrix} \alpha_t & 0 & 0 \\ 0 & \alpha_t & 0 \end{bmatrix} \quad . \tag{6.70}$$

If the object maintains an approximately constant distance from the camera, then reconstruction can be achieved by assuming constant scale $\alpha_t = 1$. If the object possesses approximately-rigid sub-structure, then rigid structure-from-motion can be used to estimate scale [68, 65]. Finally, if the camera is moving backwards and forwards or zooming, then scale may simply be estimated from the background.

Once camera scale is known, the length of each edge in the articulation graph can be estimated by its maximum observed projection [65]

$$d_{ij} = \max_t \frac{\|w_j[t] - w_i[t]\|}{\alpha_t} \quad . \tag{6.71}$$

This is a reasonable estimate due to the slow decay of the cosine function at the origin $\cos\theta \approx 1$ for small θ. This also ensures that the articulation and projection constraints will be feasible. If some edges are known to have equal length, then the maximum over several edges can be used to improve the estimate.

Reconstructions are presented in Figures 6.18, 6.19, 6.20, 6.21 and 6.22 for several real videos using the calibration-less approach. These figures are not the sole work of the author and were generated in collaboration with Yingying Zhu in the course of a joint publication. All point correspondences were

Figure 6.18: The reconstruction of several frames of a sequence from the movie "Run Lola Run." Camera estimation would be difficult as significant perspective effects are only observed for a handful of frames. Reconstruction is shown from two novel views. The human skeleton comprises 18 joints.

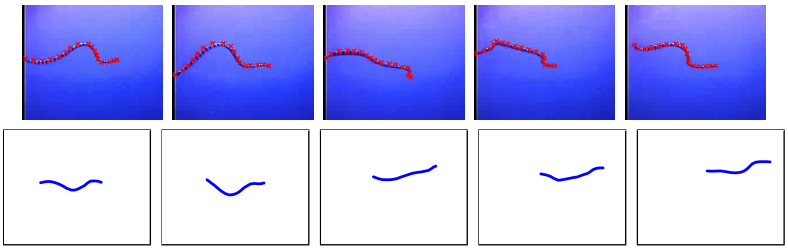


Figure 6.19: Reconstruction of a sea snake filmed underwater by a diver. Note the absence of any rigid background. The skeleton consists of 17 joints.

manually labelled to obtain these reconstructions. These sequences would be challenging or impossible cases for automatic full perspective camera estimation.

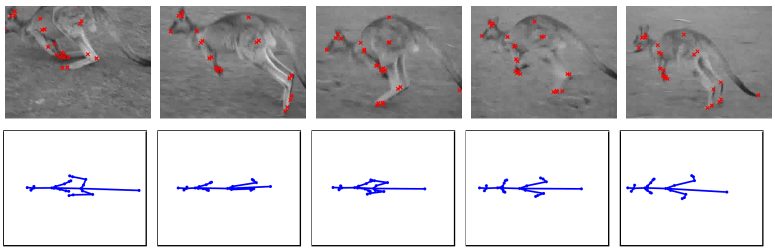


Figure 6.20: Reconstruction of a jumping kangaroo. Insufficient background texture is available to reliably estimate cameras. The skeleton consists of 23 joints.

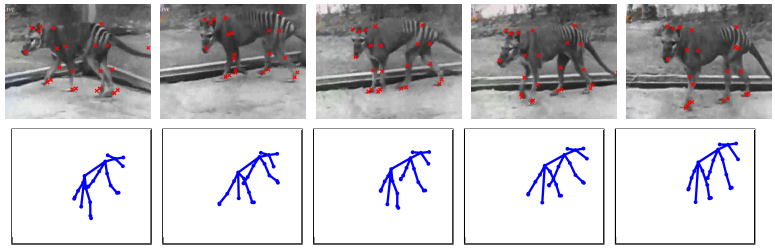


Figure 6.21: Reconstruction from the last known footage of the extinct thylacine (Tasmanian tiger). The skeleton consists of 23 joints.

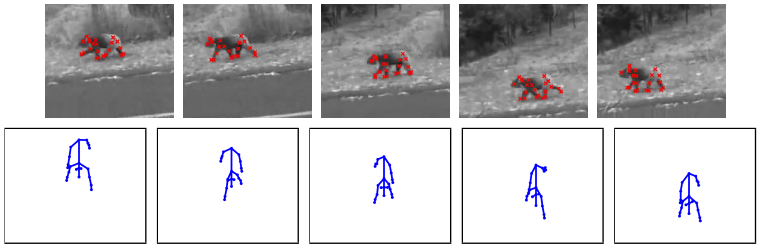


Figure 6.22: Reconstruction of a walking koala from shaky video footage, obtained following video stabilisation. The skeleton consists of 22 joints.

Chapter 7

Conclusion

7.1 Contributions

7.1.1 Object Detection

The major contribution of this work to object detection has been to compare two efficient training algorithms, Stationary Process Linear Discriminant Analysis (SPLDA) and Correlation Filters, and recognise that they can be adapted to share a common framework. Both obtain a detector by solving a linear system of equations that is defined by the covariance matrix of a large number of negative examples, however SPLDA adopts a covariance matrix that is Toeplitz whereas Correlation Filters adopt a covariance matrix that is *circulant* Toeplitz. This small difference enables the linear system in Correlation Filters (but not in SPLDA) to be constructed and solved efficiently in the Fourier domain. On the other hand, it enables the covariance matrix in SPLDA (but not in Correlation Filters) to be re-used for detectors of arbitrary size, and therefore for the linear system to be constructed without access to the negative set after this covariance matrix has been esti-

mated once. The novel framework enables the linear system in Correlation Filters to similarly be estimated for detectors of arbitrary size without access to the negative set using the non-circulant covariance matrix of SPLDA. It also demonstrates that the Fast Fourier Transform (FFT) can be utilised to efficiently estimate the non-circulant covariance matrix and, in certain situations, to solve the non-circulant system of equations.

The two algorithms were compared for the first time in a standardised framework for pedestrian detection. The results confirm that both methods approach the performance of training a Support Vector Machine (SVM) using multiple rounds of Hard Negative Mining (HNM). These experiments revealed that the assumption of a circulant covariance matrix can have a slightly detrimental effect. Timing experiments confirm that it is generally much faster to train a detector using a circulant covariance matrix than a Toeplitz covariance matrix. Iterative algorithms are proposed for solving Toeplitz linear systems, and these are demonstrated to be faster than the naive approach for large systems.

These theoretical and empirical contributions are of interest to any application that employs either of the original algorithms. This includes

- search engines where a dataset of images is queried using a single image by training a detector with one positive example,

- adaptive tracking where a detector is trained using positive examples extracted from the previous frames [6, 32],

- correspondence problems such as optical flow, stereo matching and scene flow [39], where the quality of an appearance match could be measured by training a detector per keypoint, and

- meta-algorithms such as Ensemble of Exemplar SVMs [42] that comprise many linear detectors.

7.1.2 Non-Rigid Structure-from-Motion

The major contribution of this work to Non-Rigid Structure-from-Motion (NRSfM) has been the development of a theoretical bound that better characterises the limitations of trajectory reconstruction using only temporal relationships between variables. The bound considers the reconstruction error when estimating the most likely trajectory under the assumption that trajectories are drawn from a known Gaussian distribution. The form of the bound highlights the importance of the condition number of the matrix that must be factorised to obtain a solution. Critically, it captures the effect that, when the precision matrix of the Gaussian distribution is specified using a trajectory subspace, the reconstruction error may be large if the subspace dimension is chosen too small or too large. This effect is borne out in experiments.

To find alternative precision matrices, this work introduced the assumption that the trajectory is a stationary process and therefore the precision matrix should be approximately Toeplitz. Toeplitz matrices that measure the response of the trajectory to a high-pass filter were investigated, and it was subsequently shown that the assumption of stationarity already resided in the use of trajectory subspaces defined by the Discrete Cosine Transform (DCT) due to its connection to convolution. The precision matrices that correspond to first- and second-difference filters, measuring the velocity and acceleration of a trajectory respectively, were found to have desirable spectral properties. Reconstruction using these filters avoids failure due to a poorly conditioned system without the need to manually specify a parameter such

as the subspace dimension.

Unfortunately, the accuracy with which a trajectory can be reconstructed using Gaussian temporal prior alone remains poor for realistic camera motion. However, the use of high-pass filters is still practically relevant to the problem of reconstructing a dynamic scene observed by multiple cameras [70]. Furthermore, the bound developed in this work has since been used to motivate the use of sparse coding to circumvent the problems associated with the condition number [71].

Finally, this work recognised that the structure in an objective function that computes convolution with compact filters permits an efficient solution for combinatorial problems, where the trajectory is constrained to a finite set. Although this corresponds to a graphical model that does not have a tree structure, it can still be solved efficiently using a generalisation of the max-sum algorithm. Whereas a previous approach adopted branch and bound, whose worst-case running time is still exponential in the length of the sequence, the efficient algorithm has worst-case running time that is exponential in the support of the filter and only *linear* in the length of the sequence. It is demonstrated that this method can be applied greedily to reconstruct the motion of an animal's skeleton using manually annotated keypoints, even without calibrated cameras.

7.2 Future Work

7.2.1 Object Detection

Motivated by the property that a stationary process defined on the integers has a precision matrix that is Toeplitz, it may be possible to find a banded Toeplitz approximation to the precision matrix of a finite stationary process.

This would enable detectors to be trained even more cheaply than solving a circulant system of equations, using a simple Toeplitz matrix-vector product $w = \Lambda r$. This Toeplitz approximation might be computable using an analytical expression, or perhaps it could be obtained by solving a constrained optimisation problem since the set of Toeplitz matrices is convex. It may also be possible to obtain a Toeplitz approximation to the matrix square-root $V = S^{-\frac{1}{2}}$, which could enable all patches of a window to be whitened using convolution.

Algorithms that achieve state-of-the-art performance in object detection no longer use HOG features but deep Convolutional Networks (Conv-Nets) that are trained for image classification from millions of examples using Stochastic Gradient Descent (SGD) [36]. These models typically have eight or more linear layers with learnt parameters, with at least the first five being convolutional [56, 58]. Conv-Nets can take days to train from scratch, but once a model has been learnt, it has been shown [19, 52] that its internal representation is a highly effective non-linear feature transform that generalises to many different problems. While most works that operate in this manner have adopted the output of a fully-connected layer to maximise the amount of invariance that is captured by the feature transform, some have instead taken the output of the convolutional layers as a feature transform [23, 55, 53], enabling the use of training algorithms that leverage the signal structure, such as a Deformable Parts Model [21]. The approach of using LDA with a covariance matrix estimated offline from a set of natural images has already been applied to Conv-Net features by Desai et al. [16] for image classification. However, they only considered fully-connected layers, meaning that the covariance matrix did not retain Toeplitz structure. It would be interesting to instead investigate the application of Stationary

Process LDA to the convolutional feature maps defined by the convolutional layers that precede the fully-connected layers. This may be useful for applications that require less invariance than is encoded in the fully-connected layers, such as object tracking in video, facial landmark fitting and correspondence for 3D reconstruction.

A recent approach for training Conv-Nets has proposed the use of batch normalisation [33] to prevent a change in one layer drastically affecting the inputs to the following layers. This entails normalising each output element of a layer independently to have zero mean and unit variance using element-wise statistics estimated from the examples in the SGD mini-batch. Batch normalisation enables a higher learning rate to be used, which significantly reduces training time. It could be possible to normalise adjacent elements jointly by estimating and then inverting a Toeplitz covariance matrix for each mini-batch. This may further increase the speed with which a Conv-Net can be trained.

Some of the best methods for pedestrian detection employ ensembles of decision trees [18]. The leaf nodes of a decision tree usually simply return a distribution of classes or the mode of this distribution. The ability to rapidly train a detector could be used to place a classifier instead of a constant at the leaf nodes of a decision tree to improve performance.

The generative nature of LDA could perhaps be harnessed in approaches that use an ensemble of exemplar classifiers [42] to avoid the need to calibrate the classifier outputs. Whereas the outputs of SVMs trained independently are not necessarily comparable, LDA could be used to produce likelihood estimates that can be directly compared. Classifier calibration can have a large effect on the performance of a system and is a non-trivial problem to solve [4, 45].

7.2.2 Non-Rigid Structure-from-Motion

The theoretical and experimental results in this thesis indicate that stationary Gaussian temporal prior in itself is insufficient to obtain an accurate reconstruction from monocular video in most realistic scenarios. This is strong evidence that more complex solutions are necessary, such as Kernel NRSfM [27], nuclear norm minimisation [14] and compressive sensing [71]. However, stationary Gaussian temporal prior can still be incorporated into these methods, and this work has highlighted the advantage of using compact convolutional prior over subspace projections.

7.3 Final Remarks

Signals are ubiquitous in computer vision, and this thesis has demonstrated that the tools of classical digital signal processing are useful in the analysis and design of modern solutions to a diverse set of problems.

Appendix A

Extended Derivations: Object Detection

A.1 Centroid removal in Multi-Channel Correlation Filters

The mean image of all shifts of all base examples is a uniform image which takes everywhere the mean pixel $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_k) \in \mathbb{R}^k$

$$\left(\frac{1}{mn} \sum_{i=1}^{n} \sum_{\tau \in \mathcal{U}} L_\tau x_i \right) [u] = \frac{1}{mn} \sum_{i=1}^{n} \sum_{\tau \in \mathcal{U}} x_i[\tau] = \bar{x} \ . \tag{A.1}$$

The modified base example loss is

$$\left\| \sum_{p=1}^{k} w_p \star (x_{ip} - \bar{x}_p 1) - (y_i - \bar{y} 1) \right\|^2 \tag{A.2}$$

and the resulting changes to s_{pq} and r_p are

$$s_{pq} = \frac{1}{mn} \sum_{i=1}^{n} (x_{iq} - \bar{x}_q 1) \star (x_{ip} - \bar{x}_p 1) = \frac{1}{mn} \sum_{i=1}^{n} x_{iq} \star x_{ip} - \bar{x}_p \bar{x}_q 1 \ , \tag{A.3}$$

$$r_p = \frac{1}{mn} \sum_{i=1}^{n} (y_i - \bar{y} 1) \star (x_{ip} - \bar{x}_p 1) \quad = \frac{1}{mn} \sum_{i=1}^{n} y_i \star x_{ip} - \bar{y} \bar{x}_p 1 \ . \tag{A.4}$$

with corresponding Fourier forms

$$\hat{s}_{pq}[u] = \begin{cases} \frac{1}{mn}\sum_{i=1}^{n} \hat{x}_{iq}^{*}[0]\,\hat{x}_{ip}[0] - \frac{1}{m}\hat{\bar{x}}_p[0]\,\hat{\bar{x}}_q[0] & \text{if } u = 0 \\ \frac{1}{mn}\sum_{i=1}^{n} \hat{x}_{iq}^{*}[u]\,\hat{x}_{ip}[u] & \text{if } u \neq 0 \end{cases} \tag{A.5}$$

$$\hat{r}_{p}[u] = \begin{cases} \frac{1}{mn}\sum_{i=1}^{n} \hat{y}_{i}^{*}[0]\,\hat{x}_{ip}[0] - \frac{1}{m}\hat{\bar{y}}[0]\,\hat{\bar{x}}_p[0] & \text{if } u = 0 \\ \frac{1}{mn}\sum_{i=1}^{n} \hat{y}_{i}^{*}[u]\,\hat{x}_{ip}[u] & \text{if } u \neq 0 \ . \end{cases} \tag{A.6}$$

This only additionally demands that $\hat{\bar{y}}[0] \in \mathbb{R}$ and $\hat{\bar{x}}[0] \in \mathbb{R}^k$ be accumulated and then that a cheap modification be made to the system of equations. The above expression can alternatively be given in terms of complex outer products

$$\hat{s}[u] = \begin{cases} \frac{1}{mn}\sum_{i=1}^{n} \hat{x}_{i}[0]\,\hat{x}_{i}^{H}[0] - \frac{1}{m}\hat{\bar{x}}[0]\,\hat{\bar{x}}^{T}[0] & \text{if } u = 0 \\ \frac{1}{mn}\sum_{i=1}^{n} \hat{x}_{i}[u]\,\hat{x}_{i}^{H}[u] & \text{if } u \neq 0 \ . \end{cases} \tag{A.7}$$

A.2 Equivalence of within-class and unsupervised covariance

The empirical covariance of all examples is

$$S = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T - \bar{x}\bar{x}^T \tag{A.8}$$

and the empirical within-class covariance is

$$S_W = \frac{1}{n}(n_1 S_1 + n_2 S_2) = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T - \frac{n_1}{n}\bar{x}_1\bar{x}_1^T - \frac{n_2}{n}\bar{x}_2\bar{x}_2^T \ . \tag{A.9}$$

Therefore the two are related

$$S + \bar{x}\bar{x}^T = S_W + \frac{n_1}{n}\bar{x}_1\bar{x}_1^T + \frac{n_2}{n}\bar{x}_2\bar{x}_2^T \tag{A.10}$$

which, through some manipulation, yields

$$S = S_W + \frac{n_1 n_2}{n^2}(\bar{x}_1 - \bar{x}_2)(\bar{x}_1 - \bar{x}_2)^T \ . \tag{A.11}$$

The maximum likelihood LDA solution is $w = S_W^{-1}(\bar{x}_1 - \bar{x}_2)$. If a vector x satisfies a square system of equations $Ax = b$, then $(A + kbb^T)x = (1 + kb^T x)b$ for any scalar k. Hence a solution to $(A + kbb^T)x' = b$ is $x' = ax$, where $a = 1/(1 + kb^T x)$ is positive if $k \geq 0$ and $A \succ 0$, since this implies $kb^T x = kb^T A^{-1} b \geq 0$.

Therefore the solutions using either covariance are equivalent up to a positive scalar

$$(S + \lambda I)^{-1}(\bar{x}_1 - \bar{x}_2) = a(S_W + \lambda I)^{-1}(\bar{x}_1 - \bar{x}_2) \ . \tag{A.12}$$

A.3 Least-squares regression with two labels

One expression for the right-hand side in least-squares affine regression is

$$r = \frac{1}{n}\sum_{i=n}(y_i - \bar{y})x_i \ . \tag{A.13}$$

If the labels only take two values $y_i \in \{\gamma_1, \gamma_2\}$, then the mean label is

$$\bar{y} = \frac{1}{n}(n_1\gamma_1 + n_2\gamma_2) \tag{A.14}$$

where n_1 and n_2 are the number of examples in the class defined by each label. The right-hand side r can be expressed

$$r = \frac{1}{n}\left[n_1(\gamma_1 - \bar{y})\bar{x}_1 + n_2(\gamma_2 - \bar{y})\bar{x}_2\right] \tag{A.15}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the means of each class. Using the observation that

$$n_1(\gamma_1 - \bar{y}) = \frac{n_1 n_2}{n}(\gamma_1 - \gamma_2) = -n_2(\gamma_2 - \bar{y}) \tag{A.16}$$

it is possible to express the right-hand side r in terms of the difference in means

$$r = \frac{n_1 n_2}{n^2}(\gamma_1 - \gamma_2)(\bar{x}_1 - \bar{x}_2) \ . \tag{A.17}$$

This holds for arbitrary $\gamma_1 \neq \gamma_2$.

Appendix B

Extended Derivations: Non-Rigid Structure-from-Motion

B.1 Matrix singular if nullspaces have non-trivial intersection

The matrix $Q_\perp^T \Lambda Q_\perp$ is invertible only if the nullspace of the precision matrix Λ does not intersect that of the projection matrix Q except in the trivial space

$$\text{null}(\Lambda) \cap \text{null}(Q) = \{0\} \ . \tag{B.1}$$

If the two nullspaces have a non-trivial intersection, then there exists a trajectory $x \neq 0$ such that $\Lambda x = 0$ and $x = Q_\perp z$ for some $z \neq 0$. It follows that $\Lambda Q_\perp z = 0$ and therefore $Q_\perp^T \Lambda Q_\perp$ has a nullspace and is not invertible.

B.2 Condition term is ratio of constrained optima

The condition term $\gamma(Q, \Lambda) \geq 1$ computes the condition of the precision matrix Λ for vectors confined to the nullspace of the projection matrix $\{x : Qx = 0\}$

$$\gamma(Q,\Lambda) = \frac{\lambda_{\max}(Q_\perp^T \Lambda Q_\perp)}{\lambda_{\min}(Q_\perp^T \Lambda Q_\perp)} = \left(\max_{x\neq 0, Qx=0} \frac{x^T\Lambda x}{x^T x}\right) \Big/ \left(\min_{x\neq 0, Qx=0} \frac{x^T\Lambda x}{x^T x}\right) \quad . \tag{B.2}$$

Proof. It will be shown that

$$\lambda_{\max}(Q_\perp^T \Lambda Q_\perp) = \max_{z\neq 0} \frac{z^T Q_\perp^T \Lambda Q_\perp z}{z^T z} = \max_{x\neq 0, Qx=0} \frac{x^T\Lambda x}{x^T x} \tag{B.3}$$

and then an analogous argument can be made of $\lambda_{\min}(Q_\perp^T \Lambda Q_\perp)$ to complete the proof.

By the definition of $Q_\perp$ as a basis of the nullspace of Q, there exists a *unique* $z \neq 0$ such that $x = Q_\perp z$ for any $x \neq 0$ such that $Qx = 0$. The numerators in the above expression are trivially equivalent $z^T Q_\perp^T \Lambda Q_\perp z = x^T \Lambda x$, and the denominators are equivalent $x^T x = z^T z$ due to orthonormality $Q_\perp^T Q_\perp = I$. □

B.3 Eigenvalues of semidefinite matrix under orthonormal transform

If A is an $n \times n$ symmetric positive semidefinite matrix and B is an $m \times n$ matrix with $m < n$ that satisfies $BB^T = I$, then the eigenvalues of BAB^T are a convex combination of the eigenvalues of A

$$\lambda_i(BAB^T) = \sum_{j=1}^{n} \alpha_{ij}^2 \lambda_j(A) \qquad i = 1, \ldots, m \tag{B.4}$$

where the coefficients satisfy

$$\sum_{j=1}^{n} \alpha_{ij}^2 = 1 \ . \tag{B.5}$$

Proof. Let $v_i \in \mathbb{R}^m$ be a unit eigenvector of BAB^T that satisfies $BAB^T v_i = \lambda_i(BAB^T)v_i$ and $v_i^T v_i = 1$. The corresponding eigenvalue satisfies

$$\lambda_i = v_i^T BAB^T v_i \ . \tag{B.6}$$

Let U be an $n \times n$ matrix of eigenvectors of A such that $A = U \operatorname{diag}(\lambda(A))U^T$ where $\lambda(A)$ is a vector containing the eigenvalues. There must exist $\alpha_i \in \mathbb{R}^n$ such that $B^T v_i = U\alpha_i$ since U is full rank. Let the elements of α_i be denoted $\alpha_{ij} \in \mathbb{R}$. The constraint that $v_i^T v_i = 1$ is equivalent to

$$\sum_{j=1}^{n} \alpha_{ij}^2 = 1 \tag{B.7}$$

since $\|v_i\|^2 = \|B^T v_i\|^2 = \|U\alpha_i\|^2 = \|\alpha_i\|^2$ due to orthonormality. The expression for the i-th eigenvalue is then obtained

$$\begin{aligned} \lambda_i(BAB^T) = v_i^T BAB^T v_i &= \alpha_i^T U^T AU\alpha_i \\ &= \alpha_i^T \operatorname{diag}(\lambda(A))\alpha_i = \sum_{j=1}^{n} \alpha_{ij}^2 \lambda_j(A) \ . \end{aligned} \tag{B.8}$$

□

B.4 Norm of inverse matrix monotonically increasing in basis dimension

Let $\Theta\{1, \ldots, k\}$ denote the $3\ell \times 3k$ basis for a trajectory subspace with coordinate-wise basis dimension k. It is a concatenation of the individual $3\ell \times 3$ bases $\Theta\{i\}$

$$\Theta\{1, \ldots, k\} = \begin{bmatrix} \Theta\{1\} & \cdots & \Theta\{k\} \end{bmatrix} \ . \tag{B.9}$$

Let $P_\Theta = \Theta\Theta^T$ denote the projector on to the column space of Θ. The norm of the inverse of

$$Q_\perp^T \big[I - P_{\Theta\{1,\ldots,k\}}\big] Q_\perp \tag{B.10}$$

is monotonically increasing in k

$$\left\| \left(Q_\perp^T \big[I - P_{\Theta\{1,\ldots,k\}}\big] Q_\perp\right)^{-1} \right\| \leq \left\| \left(Q_\perp^T \big[I - P_{\Theta\{1,\ldots,k+1\}}\big] Q_\perp\right)^{-1} \right\| \ . \tag{B.11}$$

Proof. The norm of the inverse of a positive-definite matrix is the inverse of its minimum eigenvalue

$$\|A^{-1}\| = \frac{1}{\lambda_{\min}(A)} \ . \tag{B.12}$$

The projection matrix for bases $\{1, \ldots, k\}$ can be written as the sum of projection matrices for each individual basis

$$P_{\Theta\{1,\ldots,k\}} = P_{\Theta\{1\}} + \cdots + P_{\Theta\{k\}} \tag{B.13}$$

and the complementary projector can be expressed

$$I - P_{\Theta\{1,\ldots,k\}} = P_{\Theta\{k+1,\ldots,\ell\}} \ . \tag{B.14}$$

The minimum eigenvalue of the sum of two positive semidefinite matrices is bounded by the minimum eigenvalues of the two matrices according to

$$\begin{aligned} \lambda_{\min}(A+B) &= \min_{x \neq 0} \frac{x^T(A+B)x}{x^T x} \\ &\geq \min_{x \neq 0} \frac{x^T A x}{x^T x} + \min_{x \neq 0} \frac{x^T B x}{x^T x} \\ &= \lambda_{\min}(A) + \lambda_{\min}(B) \end{aligned} \tag{B.15}$$

using the fundamental property that

$$\min_x [f(x) + g(x)] \geq \min_x f(x) + \min_x g(x) \ . \tag{B.16}$$

Therefore (B.11) is proved using

$$\begin{aligned}
&\lambda_{\min}\left\{Q_\perp^T\left[I - P_{\Theta\{1,\ldots,k\}}\right]Q_\perp\right\} \\
&= \lambda_{\min}\left\{Q_\perp^T\left[P_{\Theta\{k+1\}} + P_{\Theta\{k+2,\ldots,\ell\}}\right]Q_\perp\right\} \\
&\geq \lambda_{\min}\left\{Q_\perp^T P_{\Theta\{k+2,\ldots,\ell\}} Q_\perp\right\} + \lambda_{\min}\left\{Q_\perp^T P_{\Theta\{k+1\}} Q_\perp\right\} \\
&\geq \lambda_{\min}\left\{Q_\perp^T P_{\Theta\{k+2,\ldots,\ell\}} Q_\perp\right\} \\
&= \lambda_{\min}\left\{Q_\perp^T\left[I - P_{\Theta\{1,\ldots,k+1\}}\right]Q_\perp\right\} \quad . \qquad \text{(B.17)}
\end{aligned}$$

□

B.5 Expression monotonically decreasing in basis dimension

The expression $\|(I - P_\Theta)x\|$ is monotonically decreasing in the basis dimension k since the component of x that is orthogonal to the first k dimensions of an orthonormal basis $\Theta\{1,\ldots,k\}$ is at least that which is orthogonal to the first $k+1$ dimensions

$$\left\|\left(I - P_{\Theta\{1,\ldots,k\}}\right)x\right\| \geq \left\|\left(I - P_{\Theta\{1,\ldots,k+1\}}\right)x\right\| \quad . \qquad \text{(B.18)}$$

Proof. Using the properties from the previous section

$$\begin{aligned}
\left\|\left(I - P_{\Theta\{1,\ldots,k\}}\right)x\right\|^2 &= \left\|P_{\Theta\{k+1,\ldots,\ell\}}x\right\|^2 \\
&= \left\|P_{\Theta\{k+1\}}x\right\|^2 + \left\|P_{\Theta\{k+2,\ldots,\ell\}}x\right\|^2 \\
&\geq \left\|P_{\Theta\{k+2,\ldots,\ell\}}x\right\|^2 \\
&= \left\|\left(I - P_{\Theta\{1,\ldots,k+1\}}\right)x\right\|^2 \quad . \qquad \text{(B.19)}
\end{aligned}$$

□

Bibliography

[1] R. J. Adler and J. E. Taylor. *Random Fields and Geometry*. Springer Monographs in Mathematics. Springer New York, New York, NY, 2007.

[2] H. Akaike. Block Toeplitz matrix inversion. *SIAM Journal on Applied Mathematics*, 24(2):234–241, Mar. 1973.

[3] I. Akhter, Y. A. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *NIPS*. MIT Press, 2008.

[4] M. Aubry, D. Maturana, A. A. Efros, and B. C. Russell. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. pages 3762–3769, 2014.

[5] V. N. Boddeti, T. Kanade, and B. V. K. Vijaya Kumar. Correlation filters for object alignment. *CVPR*, pages 2291–2298, June 2013.

[6] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550. IEEE, June 2010.

[7] A. Böttcher and B. Silbermann. *Analysis of Toeplitz Operators*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2006.

[8] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, volume 2, pages 690–696. IEEE, 2000.

[9] H. Bristow, A. Eriksson, and S. Lucey. Fast convolutional sparse coding. In *CVPR*, pages 391–398. IEEE, June 2013.

[10] R. H. Chan and M. K. Ng. Conjugate gradient methods for Toeplitz systems. *SIAM Review*, 38(3):427–482, 1996.

[11] R. H. Chan and G. Strang. Toeplitz equations by conjugate gradients with circulant preconditioner. *SIAM Journal on Scientific and Statistical Computing*, 10(1):104–119, 1989.

[12] T. F. Chan. An optimal circulant preconditioner for Toeplitz systems. *SIAM Journal on Scientific and Statistical Computing*, 9(4):766–771, July 1988.

[13] T. F. Chan and J. A. Olkin. Circulant preconditioners for Toeplitz-block matrices. *Numerical Algorithms*, 6(1):89–101, Mar. 1994.

[14] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. In *CVPR*, 2012.

[15] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.

[16] C. Desai, J. Eledath, H. Sawhney, and M. Bansal. De-correlating CNN features for generative classification. In *WACV*, pages 428–435. IEEE, Jan. 2015.

[17] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. *BMVC*, 2009.

[18] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34(4):743–61, Apr. 2012.

[19] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for generic visual recognition. *JMLR*, 32:647–655, Oct. 2014.

[20] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[21] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, Sept. 2010.

[22] P. F. Felzenszwalb and R. Zabih. Dynamic programming and graph algorithms in computer vision. *PAMI*, 33(4):721–40, Apr. 2011.

[23] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. Technical report, UC Berkeley, Sept. 2014.

[24] I. Gohberg, S. Goldberg, and M. A. Kaashoek. *Basic Classes of Linear Operators*. Birkhäuser Basel, Basel, 2003.

[25] I. Gohberg and A. Semencul. On the inversion of finite Toeplitz matrices and their continuous analogs. *Mat. Issled.*, 2:201–233, 1972.

[26] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3 edition, 2007.

[27] P. F. U. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *ICCV*. IEEE, 2011.

[28] R. M. Gray. Toeplitz and Circulant Matrices: A Review. *Foundations and Trends in Communications and Information Theory*, 2(3):155–239, 2005.

[29] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*. Springer, 2012.

[30] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[31] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *ICCV*. IEEE, 2013.

[32] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, PP(99), Apr. 2015.

[33] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, volume 37, pages 448–456, 2015.

[34] H. Kiani Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *ICCV*. IEEE, 2013.

[35] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[37] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient Subwindow Search: A branch and bound framework for object localization. *PAMI*, 31(12):2129–2142, Dec. 2009.

[38] N. Levinson. The Wiener RMS error criterion in filter design and prediction. *Journal of Mathematics and Physics*, 1947.

[39] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011.

[40] A. Mahalanobis, B. V. K. Vijaya Kumar, and D. Casasent. Minimum average correlation energy filters. *Applied Optics*, 26(17):3633–40, Sept. 1987.

[41] A. Mahalanobis, B. V. K. Vijaya Kumar, S. Song, S. R. F. Sims, and J. F. Epperson. Unconstrained correlation filters. *Applied Optics*, 33(17):3751–9, June 1994.

[42] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of Exemplar-SVMs for object detection and beyond. In *ICCV*, pages 89–96, 2011.

[43] S. G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 3 edition, 2008.

[44] S. A. Martucci. Symmetric convolution and the discrete sine and cosine transforms. *IEEE Transactions on Signal Processing*, 42(5):1038–1051, May 1994.

[45] D. Modolo, A. Vezhnevets, O. Russakovsky, and V. Ferrari. Joint calibration of Ensemble of Exemplar SVMs. 2015.

[46] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[47] S. Nowozin and C. H. Lampert. Structured Learning and Prediction in Computer Vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011.

[48] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, 2 edition, 1999.

[49] H. S. Park and Y. A. Sheikh. 3D reconstruction of a smooth articulated trajectory from a monocular image sequence. In *ICCV*, pages 201–208. IEEE, Nov. 2011.

[50] H. S. Park, T. Shiratori, I. Matthews, and Y. A. Sheikh. 3D reconstruction of a moving point from a series of 2D projections. In *ECCV*, volume 6313, pages 158–171. Springer, 2010.

[51] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

[52] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. *CVPR DeepVision Workshop*, Mar. 2014.

[53] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. 2015.

[54] M. Salzmann and R. Urtasun. Physically-based motion models for 3D tracking: A convex formulation. In *ICCV*, pages 2064–2071. IEEE, Nov. 2011.

[55] P.-A. Savalle, S. Tsogkas, G. Papandreou, and I. Kokkinos. Deformable Part Models with CNN Features. In *ECCV Workshop on Parts and Attributes*, 2014.

[56] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, Dec. 2014.

[57] S. Serra Capizzano and E. E. Tyrtyshnikov. Any circulant-like preconditioner for multilevel matrices is not superlinear. *SIAM Journal on Matrix Analysis and Applications*, 21(2):431–439, Jan. 2000.

[58] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[59] G. Strang. A proposal for Toeplitz matrix calculations. *Studies in Applied Mathematics*, 1986.

[60] G. Strang. *Introduction to Linear Algebra*. Wellesley Cambridge Press, 4 edition, 2009.

[61] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137–154, Nov. 1992.

[62] W. F. Trench. An algorithm for the inversion of finite Toeplitz matrices. *Journal of the Society for Industrial and Applied Mathematics*, 12(3):515–522, 1964.

[63] C. K. Turnes, D. Balcan, and J. Romberg. Image deconvolution via superfast inversion of a class of two-level Toeplitz matrices. In *ICIP*, pages 3073–3076. IEEE, Sept. 2012.

[64] E. E. Tyrtyshnikov. Optimal and superoptimal circulant preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 13(2):459–473, Apr. 1992.

[65] J. Valmadre and S. Lucey. Deterministic 3D human pose estimation using rigid structure. In *ECCV*, pages 467–480. Springer, 2010.

[66] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *PAMI*, 34(3):480–492, 2012.

[67] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages 511–518. IEEE, 2001.

[68] X. Wei and J. Chai. Modeling 3D human poses from uncalibrated monocular images. In *ICCV*. IEEE, 2009.

[69] A. E. Yagle. A fast algorithm for Toeplitz-block-Toeplitz linear systems. In *ICASSP*, volume 3, pages 1929–1932. IEEE, 2001.

[70] A. Zaheer, I. Akhter, M. H. Baig, S. Marzban, and S. Khan. Multiview structure from motion in trajectory space. In *ICCV*, pages 2447–2453. IEEE, Nov. 2011.

[71] Y. Zhu and S. Lucey. Convolutional sparse coding for trajectory reconstruction. *PAMI*, 37(3):529–540, Mar. 2015.